



TUGAS AKHIR - KI141502

SISTEM PENDOKUMENTASI PENCAPAIAN PRESTASI UNTUK PEMETAAN FOKUS PEMBINAAN PERLOMBAAN BERDASARKAN ALTERNATIF PRESTASI TERBAIK

Ika Astutik
NRP 5110 100 147

Dosen Pembimbing
Sarwosri, S.Kom., M.T
Abdul Munif, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - KI141502

ACHIEVEMENT DOCUMENTATION SYSTEM FOR FOCUSED COMPETITION TRAINING MAPPING BASED ON BEST ACHIEVEMENT ALTERNATIVE

Ika Astutik
NRP 5110 100 147

Advisor
Sarwosri, S.Kom., M.T.
Abdul Munif, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

LEMBAR PENGESAHAN

**Sistem Pendokumentasi Pencapaian Prestasi untuk
Pemetaan Fokus Pembinaan Perlombaan Berdasarkan
Alternatif Prestasi Terbaik**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:
IKA ASTUTIK
NRP : 5110 100 147

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Sarwosri, S.Kom, M.T.
NIP: 19760809 200112 2 001

Abdul Munif, S.Kom, M.Sc.
NIP: 051100114



**SURABAYA
JANUARI 2015**

Sistem Pendokumentasian Pencapaian Prestasi untuk Pemetaan Fokus Pembinaan Perlombaan Berdasarkan Alternatif Prestasi Terbaik

Nama Mahasiswa : Ika Astutik
NRP : 5110 100 147
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Sarwosri, S.Kom, M.T.
Dosen Pembimbing 2 : Abdul Munif, S.Kom, M.Sc.

ABSTRAKSI

Teknik Informatika Institut Teknologi Sepuluh Nopember selalu berupaya untuk memfasilitasi mahasiswanya mencapai prestasi terbaik. Salah satu upaya yang dilakukan adalah melakukan dokumentasi terhadap prestasi-prestasi mahasiswanya. Dokumentasi masih menggunakan spreadsheet pada Google Drive dan grup Facebook, hal ini tidak efektif jika harus memuat data prestasi yang semakin besar dari tahun ke tahun.

Dengan adanya permasalahan tersebut, dalam Tugas Akhir ini dibuat sebuah sistem informasi berbasis website untuk dokumentasi prestasi mahasiswa. Data prestasi yang telah terdokumentasi digunakan untuk mendapatkan rekomendasi prestasi dan rekomendasi mahasiswa. Rekomendasi prestasi diperoleh dengan metode Fuzzy Multi-Attribute Decision Making Simple Additive Weighting (SAW) dan rekomendasi mahasiswa diperoleh dengan memetakan cabang kegiatan dengan rumpun mata kuliah.

Rekomendasi prestasi diharapkan mampu membantu dalam penentuan pemilihan prioritas keikutsertaan cabang kegiatan dengan mempertimbangkan peringkat rekomendasi prestasi terbaik. Sedangkan rekomendasi mahasiswa

digunakan untuk memperoleh daftar nama mahasiswa yang direkomendasikan mengikuti suatu cabang kegiatan.

Kata kunci: *Dokumentasi Prestasi, Fuzzy Multi-Attribute Decision Making Simple Additive Weighting, Pemetaan, Rekomendasi Mahasiswa, Rekomendasi Prestasi.*

Achievement Documentation System for Focused Competition Training Mapping Based on Best Achievement Alternative

Student Name : Ika Astutik
NRP : 51010 100 147
Major : Teknik Informatika FTIf-ITS
Advisor 1 : Sarwosri, S.Kom, M.T.
Advisor 2 : Abdul Munif, S.Kom, M.Sc.

ABSTRACT

Information Engineering Institute of Technology strives to facilitate the students achieve the best performance. One of the efforts is to document to the achievements of students. Documentation is still using spreadsheets in Google Drive and Facebook group, it is not effective if it should contain greater achievement data from year to year.

Given these problems, in this final project created a web-based information system for documentation of student achievement. Achievement data that has been documented is used to obtain recommendations student achievement and recommendations. Recommendations achievement obtained with the method of Fuzzy Multi-Attribute Decision Making Simple Additive Weighting (SAW) and the recommendations obtained by mapping the student branch of activity with clumps of course.

Recommendations achievement is expected to help in the selection of priority branches of activity participation by considering the recommendation rank the best performance. While the recommendation of students used to obtain a list of names of students are recommended to follow a branch of activity.

Keywords: Documentation Achievement, Fuzzy Multi-Attribute Decision Making Simple Additive Weighting, Mapping, Recommendations Achievement, Recommendations Students.

KATA PENGANTAR

Puji syukur Alhamdulillah kepada Allah Yang Maha Kuasa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“Sistem Pendokumentasi Pencapaian Prestasi untuk Pemetaan Fokus Pembinaan Perlombaan Berdasarkan Alternatif Prestasi Terbaik”

Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Ibuk Mudrikah, Bapak Sarbianto, adik Diana Ernawati, dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Ibu Sarwosri dan Bapak Abdul Munif selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan bimbingan selama proses pengerjaan Tugas Akhir ini.
3. Bapak, Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Teman-teman seperjuangan, Anita, Awalia, Muamar, Aminudin, Nada, Fadjar, Hani, Ardian, Hardika, Fahmi (yang sudah S.Kom duluan), Miranti, Aida, Linda, Orysa, Jarwe, Rizka, Septi dan Luluk yang selama ini luar biasa.
6. Teman-teman senior, mas rohmad, mas misbachul, mbak yola, mbak nur, mbak dian, mbak dayu, mbak adel yang selalu mencairkan kejenuhan dan membantu

penulis ketika kesulitan dalam proses penyusunan Tugas Akhir ini.

7. Teman-teman angkatan 2010 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama 4 tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan Tugas Akhir ini.
8. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu-persatu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Januari 2015

Ika Astutik

DAFTAR ISI

LEMBAR PENGESAHAN	vii
ABSTRAKSI	ix
ABSTRACTION.....	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR PERSAMAAN	xxxiii
DAFTAR TABEL.....	xxvii
DAFTAR KODE SUMBER	xxxiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	4
1.3. Rumusan Permasalahan	4
1.4. Batasan Permasalahan.....	4
1.5. Metodologi	5
1.6. Sistematika Penulisan	6
BAB II DASAR TEORI	9
2.1. Sistem Dokumentasi Prestasi Jurusan Teknik Informatika.....	9
2.2. Fuzzy Multi-Attribute Decision Making Simple Additive Weighting.....	9
2.2.1. <i>Simple Additive Weighting (SAW)</i>	9
2.3. MVC (Model, View, Controller) Pattern	11

2.4.	ASP.NET	12
2.5.	Bootstrap	12
2.6.	ASP.NET MVC 4	13
2.7.	ADO.NET Entity Framework 5	14
BAB III ANALISIS DAN PERANCANGAN SISTEM		17
3.1.	Analisis	17
3.1.1.	Analisis Permasalahan	17
3.1.2.	Deskripsi Umum Perangkat Lunak	18
3.1.3.	Spesifikasi Kebutuhan Perangkat Lunak	18
3.2.	Perancangan Sistem	33
3.2.1.	Perancangan Database.....	33
3.2.2.	Perancangan Proses Aplikasi	44
3.2.3.	Perancangan Kelas	52
3.2.4.	Perancangan Arsitektur Sistem	52
3.2.5.	Perancangan Antarmuka Grafis	53
BAB IV IMPLEMENTASI		59
4.1.	Lingkungan Implementasi.....	59
4.1.1.	Lingkungan Implementasi Perangkat Keras .	59
4.1.2.	Lingkungan Implementasi Perangkat Lunak	59
4.2.	Implementasi Proses Aplikasi.....	59
4.2.1.	Implementasi Model	59
4.2.2.	Implementasi View	61
4.2.3.	<i>Implementasi</i> Control.....	120
BAB V PENGUJIAN DAN EVALUASI		131
5.1.	Lingkungan Pengujian	131

5.2.	Skenario Pengujian	131
5.2.1.	Pengujian Rekomendasi <i>Simple Additive Weighting</i> Dengan <i>Microsoft Excel</i>	131
5.2.2.	Pengujian Subjektif.....	135
5.2.3.	Pengujian Fungsionalitas	148
5.2.4.	Pengujian Verifikasi Hasil Rekomendasi....	180
BAB VI KESIMPULAN DAN SARAN		187
6.1.	Kesimpulan	187
6.2.	Saran	187
DAFTAR PUSTAKA		189
LAMPIRAN A. ANALISIS DAN PERANCANGAN SISTEM		191
8.1.	Skenario Kasus Penggunaan	191
8.2.	Diagram Aktivitas	204
8.3.	Diagram Sekuens	214
8.4.	Perancangan Database.....	221
8.5.	Perancangan Diagram Kelas	223
LAMPIRAN B. KODE SUMBER.....		229
LAMPIRAN C. PENGUJIAN		243
BIODATA PENULIS		249

DAFTAR TABEL

Tabel 3.1 Daftar Kode Kebutuhan Fungsional Sistem.....	19
Tabel 3.2 Daftar Kode Diagram Kasus Penggunaan	21
Tabel 3.3 Skenario Mengelola Data User	22
Tabel 3.4 Skenario Melihat Laporan Prestasi	26
Tabel 3.5 Penjelasan Tabel Anggota.....	34
Tabel 3.6 Penjelasan Tabel Berita.....	34
Tabel 3.7 Penjelasan Tabel Cabang Kegiatan.....	34
Tabel 3.8 Penjelasan Tabel Dosbing.....	35
Tabel 3.9 Penjelasan Tabel Dosen	35
Tabel 3.10 Penjelasan Tabel History Prestasi	35
Tabel 3.11 Penjelasan Tabel Jenis Kegiatan	36
Tabel 3.12 Penjelasan Tabel Jenis Nilai	36
Tabel 3.13 Penjelasan Tabel Kegiatan Prestasi.....	36
Tabel 3.14 Penjelasan Tabel Kriteria.....	37
Tabel 3.15 Penjelasan Tabel Mahasiswa	37
Tabel 3.16 Penjelasan Tabel Mata Kuliah	37
Tabel 3.17 Penjelasan Tabel Nilai Mahasiswa	38
Tabel 3.18 Penjelasan Tabel Nilai <i>Range</i>	38
Tabel 3.19 Penjelasan Tabel Pengguna.....	39
Tabel 3.20 Penjelasan Tabel Peringkat Kegiatan.....	39
Tabel 3.21 Penjelasan Tabel Permohonan Dana.....	39
Tabel 3.22 Penjelasan Tabel Prestasi	40
Tabel 3.23 Penjelasan Tabel RMK	40
Tabel 3.24 Penjelasan Tabel RMK Lomba	41
Tabel 3.25 Penjelasan Tabel role	41
Tabel 3.26 Penjelasan Tabel Status Berita	41
Tabel 3.27 Penjelasan Tabel Status Mahasiswa.....	42
Tabel 3.28 Penjelasan Tabel Status Permohonan	42
Tabel 3.29 Penjelasan Tabel Status Prestasi	42
Tabel 3.30 Penjelasan Tabel Tingkat Kegiatan.....	42
Tabel 3.31 Penjelasan Tabel Variabel.....	43
Tabel 3.32 Penjelasan Tabel Rekomendasi SAW	43

Tabel 3.33 Contoh Data Prestasi Mahasiswa Teknik Informatika.....	44
Tabel 3.34 Contoh Alternatif dari Data Prestasi Teknik Informatika.....	45
Tabel 3.35 Kriteria untuk sistem JUARA.....	45
Tabel 3.36 Jumlah Poin Juara	46
Tabel 3.37 Kriteria (C_i).....	46
Tabel 3.38 Bobot Variabel.....	47
Tabel 3.39 Kriteria dan Bobot Jumlah Peserta (C_1).....	47
Tabel 3.40 Kriteria dan Bobot Jumlah Finalis (C_2).....	47
Tabel 3.41 Kriteria dan Bobot Poin Juara (C_3).....	48
Tabel 3.42 Kriteria dan Bobot Poin Tingkat Kegiatan (C_4).....	48
Tabel 3.43 Hasil Pengecekan Bobot Setiap Kriteria pada Alternatif.....	48
Tabel 3.44 Kriteria dan Bobot Ternormalisasi.....	50
Tabel 3.45 Hasil.....	51
Tabel 3.46 Daftar Pilihan pada Setiap Bagian Pilihan Manajemen.....	54
Tabel 4.1 Daftar Kelas Model.....	60
Tabel 4.2 Daftar Region dalam Kelas DosenController	121
Tabel 5.1 Data Diri Pengisi Kuisioner (Mahasiswa).....	137
Tabel 5.2 Data Diri Pengisi Kuisioner (Dosen)	137
Tabel 5.3 Hasil Kuisioner Penggunaan Website Juara oleh Mahasiswa.....	138
Tabel 5.4 Presentase Alternatif Jawaban Pada Kuisioner Mahasiswa.....	138
Tabel 5.5 Presentase Interpretasi Penilaian Kuisioner Mahasiswa.....	139
Tabel 5.6 Interpretasi Skor Hasil Penilaian Kuisioner Mahasiswa.....	140
Tabel 5.7 Hasil Kuisioner Penggunaan Website Juara oleh Dosen	140
Tabel 5.8 Presentase Alternatif Jawaban Pada Kuisioner Dosen	142

Tabel 5.9 Presentase Interpretasi Penilaian Kuisoner Dosen	143
Tabel 5.10 Interpretasi Skor Hasil Penilaian Kuisoner Dosen	144
Tabel 5.11 Hasil Kuisoner Penggunaan Website Juara Secara Umum	144
Tabel 5.12 Presentase Alternatif Jawaban Pada Kuisoner umum	145
Tabel 5.13 Presentase Interpretasi Penilaian Kuisoner Umum	145
Tabel 5.14 Interpretasi Skor Hasil Penilaian Kuisoner Umum	145
Tabel 5.15 Hasil Kuisoner Penilaian untuk JUARA.....	146
Tabel 5.16 Hasil Kuisoner Testimoni Atau Pendapat Tentang JUARA.....	147
Tabel 5.17 Daftar Pengujian Fungsional.....	148
Tabel 5.18. Tabel Spesifikasi Pengujian Fungsional Menambah Data Prestasi ke Dalam Sistem	149
Tabel 5.19. Tabel Spesifikasi Pengujian Fungsional Menghapus Data Prestasi ke Dalam Sistem.....	151
Tabel 5.20. Tabel Spesifikasi Pengujian Fungsional Mengubah Data Prestasi ke Dalam Sistem	153
Tabel 5.21. Tabel Spesifikasi Pengujian Fungsional Menambah Rumpun Mata Kuliah ke Dalam Sistem	155
Tabel 5.22. Tabel Spesifikasi Pengujian Fungsional Menambah Mata Kuliah ke Dalam Sistem	156
Tabel 5.23. Tabel Spesifikasi Pengujian Fungsional Menambah Jenis Nilai ke Dalam Sistem	157
Tabel 5.24. Tabel Spesifikasi Pengujian Fungsional Menambah Jenis Nilai ke Dalam Sistem	158
Tabel 5.25. Tabel Spesifikasi Pengujian Fungsional Melihat Rekomendasi Mahasiswa ke Dalam Sistem.....	160
Tabel 5.26. Tabel Spesifikasi Pengujian Fungsional Menambah Kriteria ke Dalam Sistem.....	161

Tabel 5.27. Tabel Spesifikasi Pengujian Fungsional Menambah Variabel ke Dalam Sistem	162
Tabel 5.28. Tabel Spesifikasi Pengujian Fungsional Menambah Bobot Kriteria ke Dalam Sistem	164
Tabel 5.29. Tabel Spesifikasi Pengujian Fungsional Melihat Rekomendasi Mahasiswa ke Dalam Sistem.....	166
Tabel 5.30. Tabel Spesifikasi Pengujian Fungsional Mengubah Permohonan Dana ke Dalam Sistem.....	167
Tabel 5.31. Tabel Spesifikasi Pengujian Fungsional Menambah Berita ke Dalam Sistem	169
Tabel 5.32. Tabel Spesifikasi Pengujian Fungsional Mengubah Berita ke Dalam Sistem	171
Tabel 5.33. Tabel Spesifikasi Pengujian Fungsional Menghapus Berita ke Dalam Sistem.....	173
Tabel 5.34. Tabel Spesifikasi Pengujian Fungsional Menambah Info Prestasi ke Dalam Sistem	174
Tabel 5.35. Tabel Spesifikasi Pengujian Fungsional Menambah Permohonan Dana ke Dalam Sistem.....	176
Tabel 5.36. Tabel Spesifikasi Pengujian Fungsional Menambah Nilai ke Dalam Sistem	177
Tabel 5.37. Tabel Spesifikasi Pengujian Fungsional Menambah Berita ke Dalam Sistem	179
Tabel 5.38. Pertanyaan Kuisioner untuk Pengujian Verifikasi	184
Tabel 5.39. Tabel Hasil Kuisioner untuk Pengujian Verifikasi	184
Tabel 5.40. Presentase Alternatif Jawaban Pada Kuisioner Pengujian Verifikasi.....	184
Tabel 5.41. Presentase Interpretasi Penilaian Kuisioner Verifikasi	185
Tabel 5.42. Interpretasi Skor Hasil Penilaian Kuisioner Verifikasi	185
Tabel 8.1. Skenario Mengelola Data Prestasi	191
Tabel 8.2. Skenario Mengelola Data Cabang Kegiatan	193
Tabel 8.3. Skenario Mengelola Permohonan Dana.....	194

Tabel 8.4 Skenario Melihat Rekomendasi Mahasiswa	196
Tabel 8.5 Skenario Melihat Rekomendasi Prestasi.....	197
Tabel 8.6 Skenario Mengelola Berita	197
Tabel 8.7 Skenario Mengunggah Info Prestasi	198
Tabel 8.8 Skenario Mengunggah Nilai	200
Tabel 8.9 Skenario Mengunggah Permohonan Dana.....	201
Tabel 8.10 Skenario Mengelola Berita	202

DAFTAR GAMBAR

Gambar 1.1 Dokumentasi Prestasi Menggunakan Spreadsheet	2
Gambar 1.2 Grup IFRITS Winning Team	3
Gambar 2.1 Arsitektur Model-View-Controller [3]	11
Gambar 2.2 Interaksi Aplikasi MVC [5]	13
Gambar 2.3 Arsitektur Entity Framework [6]	14
Gambar 3.1 Diagram Kasus Penggunaan untuk Aktor Admin	20
Gambar 3.2 Diagram Kasus Penggunaan untuk Aktor Mahasiswa	21
Gambar 3.3 Diagram Aktivitas Kasus Penggunaan Mengelola User	23
Gambar 3.4 Diagram Sekuens dari Kasus Penggunaan Mengelola User	24
Gambar 3.5 Diagram Aktivitas Melihat Laporan Prestasi	27
Gambar 3.6 Diagram Sekuens Melihat Laporan Prestasi	28
Gambar 3.7 Rancangan Arsitektur Sistem JUARA	53
Gambar 3.8 Gambar Halaman Utama Juara	54
Gambar 3.9 Rancangan Form	55
Gambar 3.10 Rancangan Dashboard JUARA untuk Admin	56
Gambar 3.11 Halaman Dashboard Admin Berita	56
Gambar 3.12 Halaman Dashboard Admin Permohonan Dana	57
Gambar 3.13 Halaman Dashboard Mahasiswa	58
Gambar 4.1 Antarmuka Halaman Utama	63
Gambar 4.2 Tampilan Berita Pada Halaman Home	63
Gambar 4.3 Antarmuka Halaman Utama Dashboard Dosen	64
Gambar 4.4 Antarmuka Halaman Daftar Pengguna	65
Gambar 4.5 Form untuk Menambah Data Pengguna	66
Gambar 4.6 Form untuk Edit Data Pengguna	66
Gambar 4.7 Isi Halaman Data Mahasiswa	67
Gambar 4.8 Form untuk Menambah Data Mahasiswa	67
Gambar 4.9 Form untuk Edit Data Mahasiswa	68

Gambar 4.10 Antarmuka Halaman Data Dosen.....	68
Gambar 4.11 Form untuk Menambah Data Dosen	69
Gambar 4.12 Form untuk Edit Data Pengguna	69
Gambar 4.13 Antarmuka Halaman Data Prestasi	70
Gambar 4.14 Form untuk Menambah Data Prestasi	71
Gambar 4.15 Form untuk Edit Data Prestasi	72
Gambar 4.16 Antarmuka Halaman Tambah Anggota.....	73
Gambar 4.17 Form untuk Tambah Anggota	73
Gambar 4.18 Antarmuka Halaman Tambah Dosen	74
Gambar 4.19 Form untuk Tambah Dosen.....	74
Gambar 4.20 Antarmuka Halaman Histori Prestasi.....	75
Gambar 4.21 Form untuk Update Status Prestasi	75
Gambar 4.22 Antarmuka Halaman Form Edit Info Prestasi dengan Peringkat.....	76
Gambar 4.23 Antarmuka Halaman Data Peringkat Kejuaraan	77
Gambar 4.24 Form untuk Menambah Data Peringkat Kejuaraan	77
Gambar 4.25 Form untuk Edit Data Pengguna	78
Gambar 4.26 Antarmuka Halaman Data Jenis Kegiatan	78
Gambar 4.27 Form untuk Menambah Data Jenis Kegiatan ..	79
Gambar 4.28 Form untuk Edit Data Jenis Kegiatan	79
Gambar 4.29 Antarmuka Halaman Data Tingkat Kegiatan..	80
Gambar 4.30 Form untuk Menambah Data Tingkat Kegiatan	80
Gambar 4.31 Form untuk Edit Data Tingkat Kegiatan	81
Gambar 4.32 Antarmuka Halaman Data Kegiatan	81
Gambar 4.33 Form untuk Menambah Data Kegiatan	82
Gambar 4.34 Form untuk Edit Data Kegiatan	82
Gambar 4.35 Antarmuka Halaman Data Cabang Kegiatan ..	83
Gambar 4.36 Form untuk Menambah Data Cabang Kegiatan	83
Gambar 4.37 Form untuk Edit Data Cabang Kegiatan	84
Gambar 4.38 Antarmuka Halaman Permohonan Dana.....	85
Gambar 4.39 Form untuk Menambah Permohonan Dana	85

Gambar 4.40 Form untuk Edit Permohonan Dana.....	86
Gambar 4.41 Antarmuka Grafik Prestasi	87
Gambar 4.42 Laporan Prestasi	87
Gambar 4.43 Antarmuka Halaman Data Rumpun Mata Kuliah	88
Gambar 4.44 Form untuk Menambah Data Rumpun Mata Kuliah.....	88
Gambar 4.45 Form untuk Edit Data Rumpun Mata Kuliah..	89
Gambar 4.46 Antarmuka Halaman Data Mata Kuliah.....	90
Gambar 4.47 Form untuk Menambah Data Mata Kuliah	90
Gambar 4.48 Form untuk Edit Data Mata Kuliah.....	91
Gambar 4.49 Antarmuka Halaman Data Jenis Nilai.....	91
Gambar 4.50 Form untuk Menambah Data Jenis Nilai.....	92
Gambar 4.51 Form untuk Edit Data Jenis Nilai.....	92
Gambar 4.52 Antarmuka Halaman Pemetaan.....	93
Gambar 4.53 Antarmuka Halaman Tambah Pemetaan.....	93
Gambar 4.54 Tabel Daftar RMK	94
Gambar 4.55 Form Tambah Pemetaan	94
Gambar 4.56 Antarmuka Halaman Rekomendasi Mahasiswa	95
Gambar 4.57 Antarmuka Halaman Kriteria	95
Gambar 4.58 Form untuk Menambah Kriteria.....	96
Gambar 4.59 Form untuk Edit Kriteria.....	96
Gambar 4.60 Antarmuka Halaman Variabel.....	97
Gambar 4.61 Form untuk Menambah Variabel	97
Gambar 4.62 Form untuk Edit Variabel.....	98
Gambar 4.63 Antarmuka Halaman Atur Nilai Range.....	98
Gambar 4.64 Form untuk Mengubah Nilai Range.....	99
Gambar 4.65 Form untuk Edit Nilai <i>Range</i>	99
Gambar 4.66 Antarmuka Halaman Rekomendasi Prestasi .	100
Gambar 4.67 Antarmuka Halaman Data Status Prestasi.....	101
Gambar 4.68 Form untuk Menambah Data Status Prestasi	101
Gambar 4.69 Form untuk Edit Data Status Prestasi.....	102
Gambar 4.70 Antarmuka Halaman Data Status Permohonan	102

Gambar 4.71 Form untuk Menambah Data Status Permohonan	103
Gambar 4.72 Form untuk Edit Data Status Permohonan	103
Gambar 4.73 Antarmuka Halaman Data Status Mahasiswa	104
Gambar 4.74 Form untuk Menambah Data Status Mahasiswa	104
Gambar 4.75 Form untuk Edit Data Status Mahasiswa	105
Gambar 4.76 Antarmuka Halaman Data Status Berita	105
Gambar 4.77 Form untuk Menambah Data Status Berita...	106
Gambar 4.78 Form untuk Edit Data Status Berita	106
Gambar 4.79 Antarmuka Halaman Data Role	107
Gambar 4.80 Form untuk Menambah Data Role	107
Gambar 4.81 Form untuk Edit Data <i>Role</i>	108
Gambar 4.82 Antarmuka Halaman Manajemen Berita.....	108
Gambar 4.83 Form untuk Menambah Berita	109
Gambar 4.84 Form untuk Edit Berita.....	110
Gambar 4.85 Antarmuka Halaman Info Prestasi	111
Gambar 4.86 Form untuk Menambah Info Prestasi.....	111
Gambar 4.87 Form untuk Edit Info Prestasi	112
Gambar 4.88 Antarmuka Halaman Mahasiswa Tambah Anggota.....	113
Gambar 4.89 Form untuk Tambah Anggota	113
Gambar 4.90 Antarmuka Halaman Mahasiswa Tambah Dosen	114
Gambar 4.91 Form untuk Tambah Dosen.....	114
Gambar 4.92 Antarmuka Halaman Histori Prestasi.....	114
Gambar 4.93 Form untuk Update Status Prestasi	115
Gambar 4.94 Antarmuka Halaman Form Edit Info Prestasi dengan Peringkat.....	115
Gambar 4.95 Antarmuka Halaman Permohonan Dana.....	116
Gambar 4.96 Form untuk Menambah Permohonan Dana ..	117
Gambar 4.97 Form untuk Edit Permohonan Dana.....	117
Gambar 4.98 Antarmuka Halaman Nilai Kuliah Mahasiswa	118

Gambar 4.99 Form untuk Menambah Nilai Kuliah Mahasiswa	118
Gambar 4.100 Form untuk Edit Nilai Kuliah Mahasiswa...	119
Gambar 4.101 Antarmuka Halaman Berita.....	119
Gambar 4.102 Form untuk Menambah Berita	120
Gambar 4.103 Contoh Hasil Query dari Data Sementara ...	125
Gambar 5.1 Matrix Kriteria Hasil Dari Query	133
Gambar 5.2 Matirx Keputusan Hasil Dari Query	134
Gambar 5.3 Normalisasi Matriks Keputusan.....	135
Gambar 5.4 Rangking Normalisasi Matriks Keputusan Dan Value.....	135
Gambar 5.5 Rekomendasi Prestasi Hasil Sistem Perhitungan Sistem JUARA.....	135
Gambar 5.6 Data Prestasi Tersimpan dalam Sistem	150
Gambar 5.7 Memasukan Data Prestasi	151
Gambar 5.8 Memilih Data untuk Dihapus	152
Gambar 5.9 Pesan untuk Konfirmasi Hapus Data Prestasi .	152
Gambar 5.10 Data Prestasi Tim Igem Berhasil Dihapus Sistem	152
Gambar 5.11 Memilih Data untuk Diperbarui	153
Gambar 5.12 Form untuk Mengubah Data Baru.....	154
Gambar 5.13 Hasil Mengubah Data Prestasi Berhasil	154
Gambar 5.14 Memasukan Rumpun Mata Kuliah	155
Gambar 5.15 Rumpun Mata Kuliah Tersimpan Dalam Sistem	156
Gambar 5.16 Memasukan Data Mata Kuliah.....	157
Gambar 5.17 Data Mata Kuliah Baru Tersimpan Dalam Sistem	157
Gambar 5.18 Memasukan Jenis Nilai	158
Gambar 5.19 Jenis Nilai Tersimpan dalam Sistem	158
Gambar 5.20 Memilih Cabang yang Akan Dipetakan	159
Gambar 5.21 Tambah RMK pada Cabang Kegiatan	159
Gambar 5.22 Berhasil Tambah Rumpun Mata Kuliah.....	160
Gambar 5.23 Tampilan Rekomendasi Mahasiswa.....	161
Gambar 5.24 Memasukan Kriteria.....	162

Gambar 5.25 Kriteria Baru Tersimpan Dalam Sistem	162
Gambar 5.26 Memasukan Variabel	163
Gambar 5.27 Variabel Baru Tersimpan Dalam Sistem.....	163
Gambar 5.28 Memasukan Bobot	165
Gambar 5.29 Bobot Baru Tersimpan Dalam Sistem.....	165
Gambar 5.30 Tampilan Rekomendasi Mahasiswa.....	166
Gambar 5.31 Memilih Permohonan Dana yang Diubah.....	167
Gambar 5.32 Status Permohonan Dana Sebelum Diubah...	168
Gambar 5.33 Status Permohonan Dana yang Akan Diperbarui	168
Gambar 5.34 Status Baru Permohonan Dana Tersimpan....	168
Gambar 5.35 Memasukan Berita	170
Gambar 5.36 Berita Berhasil Tersimpan.....	170
Gambar 5.37 Memasukan Data Baru Berita	172
Gambar 5.38 Berita Baru Telah Tersimpan Dalam Sistem.	172
Gambar 5.39 Memilih Data untuk Dihapus	173
Gambar 5.40 Pesan untuk Konfirmasi Hapus Data Prestasi	173
Gambar 5.41 Data Prestasi Tim Igem Berhasil Dihapus Sistem	174
Gambar 5.42 Memasukan Info Prestasi	175
Gambar 5.43 Info Prestasi Tersimpan dalam Sistem	175
Gambar 5.44 Memasukan Permohonan Dana.....	177
Gambar 5.45 Permohonan Dana Tersimpan dalam Sistem	177
Gambar 5.46 Memasukan Nilai Kuliah	178
Gambar 5.47 Nilai Kuliah Tersimpan dalam Sistem	179
Gambar 5.48 Memasukan Berita	180
Gambar 5.49 Berita Berhasil Tersimpan.....	180
Gambar 5.50 Data Prestasi Gemastik 7 Mahasiswa Teknik Informatika.....	181
Gambar 5.51 Matrix Kriteria Hasil Dari Query	182
Gambar 5.52 Matirx Keputusan Hasil Dari Query	182
Gambar 5.53 Rangking Normalisasi Matriks Keputusan Dan Value.....	183
Gambar 5.54 Rekomendasi Prestasi Hasil Sistem Perhitungan Sistem JUARA.....	183

Gambar 8.1 Digram Aktivitas Kasus Penggunaan Dokumentasi Prestasi	204
Gambar 8.2 Diagram Aktivitas Mengelola Data Cabang Kegiatan	205
Gambar 8.3 Diagram Aktivitas Kasus Penggunaan Mengelola Permohonan Dana	206
Gambar 8.4 Diagram Aktivitas Kasus Penggunaan Melihat Rekomendasi Mahasiswa	207
Gambar 8.5 Diagram Aktivitas Kasus Penggunaan Melihat Rekomendasi Prestasi	208
Gambar 8.6 Diagram Aktivitas Kasus Penggunaan Mengelola Berita	209
Gambar 8.7 Diagram Aktivitas Kasus Penggunaan Mengunggah Info Prestasi	210
Gambar 8.8 Diagram Aktivitas Kasus Penggunaan Mengunggah Nilai Kuliah	211
Gambar 8.9 Diagram Aktivitas Kasus Penggunaan Mengunggah Permohonan Dana	212
Gambar 8.10 Diagram Aktivitas Kasus Penggunaan	213
Gambar 8.11 Diagram Sekuens Mengelola Dokumentasi Prestasi	214
Gambar 8.12 Diagram Sekuens Mengelola Cabang Kegiatan	215
Gambar 8.13 Diagram Sekuens Mengelola Permohonan Dana	215
Gambar 8.14 Diagram Sekuens Rekomendasi Mahasiswa	216
Gambar 8.15 Diagram Sekuens Rekomendasi Prestasi	216
Gambar 8.16 Diagram Sekuens Mengelola Berit	216
Gambar 8.17 Diagram Sekuens Mengunggah Info Prestasi	217
Gambar 8.18 Diagram Sekuens Mengunggah Nilai	218
Gambar 8.19 Diagram Sekuens Mengunggah Permohonan Dana	218
Gambar 8.20 Diagram Sekuens Mengunggah Berita	219
Gambar 8.21 Conceptual Data Model Aplikasi Juara	221
Gambar 8.22 Physical Data Model Aplikasi Juara	222

Gambar 8.23 Diagram Kelas Model Aplikasi JUARA	223
Gambar 8.24 Diagram Kelas Kontrol Home Controller	224
Gambar 8.25 Diagram Kelas Kontrol UserController	224
Gambar 8.26 Diagram Kelas Kontrol MahasiswaController	225
Gambar 8.27 Diagram Kelas Kontrol DosenController	226
Gambar 8.28 Diagram Kelas Aplikasi JUARA	227

BAB I

PENDAHULUAN

Pada bab ini dipaparkan mengenai garis besar Tugas Akhir, meliputi latar belakang, tujuan, rumusan permasalahan, batasan permasalahan, metodologi penyelesaian Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Pendidikan nasional berfungsi mengembangkan kemampuan dan membentuk watak serta peradaban bangsa yang bermartabat dalam rangka mencerdaskan kehidupan bangsa. Pendidikan nasional bertujuan untuk mengembangkan potensi peserta didik (mahasiswa) agar menjadi manusia yang beriman dan bertakwa kepada Tuhan Yang Maha Esa, berakhlak mulia, sehat, berilmu, cakap, kreatif, mandiri, penuh inisiatif, bekerja secara cermat, penuh tanggung jawab dan gigih. Mahasiswa diharapkan tidak hanya menekuni materi yang diperoleh dari proses perkuliahan dalam kelas saja, tetapi juga mengembangkan ilmunya dengan beraktivitas di luar jam kuliah. Sebagai seorang mahasiswa diharapkan mampu melakukan aktivitas-aktivitas yang mengembangkan kemampuan teknis dan kemampuan non teknis secara seimbang, agar nantinya menjadi lulusan yang mandiri, penuh inisiatif, bekerja secara cermat, penuh tanggung jawab dan gigih. Kemampuan ini dapat diperoleh mahasiswa melalui pembekalan secara formal dalam kurikulum pembelajaran, kokurikuler, dan ekstrakurikuler [1]. Namun, tidak semua mau dan mampu untuk menjadi pembelajar yang sukses. Seringkali mahasiswa menggunakan waktunya untuk melakukan aktivitas-aktivitas yang tidak mendukung proses untuk menjadi sarjana yang siap berkompetisi. Sementara itu, dalam era persaingan bebas yang dibutuhkan lulusan yang memiliki kemampuan teknis dan kemampuan non teknis yang seimbang. Oleh karenanya di tiap perguruan tinggi perlu mempersiapkan

mahasiswanya untuk memperoleh prestasi sebanyak mungkin melalui berbagai kegiatan seperti perlombaan, konferensi, ajang penghargaan, forum dan kegiatan positif lainnya. Dengan prestasi tersebut akan mudah tergambar kualitas seorang mahasiswa, sehingga akan mudah dalam melakukan pemetaan pembinaan bidang unggulan dan pemetaan perbaikan bidang-bidang ilmu yang kurang dikuasai oleh mahasiswanya. Sehingga muncul tugas baru bagi perguruan tinggi untuk melakukan dokumentasi terhadap prestasi mahasiswanya guna mengetahui potensi-potensi mahasiswanya, evaluasi terhadap proses pembinaan, dan sebagai motivasi untuk mendorong mahasiswa lainnya meraih prestasi.

Tak terkecuali jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember, juga melakukan pendokumentasian terhadap prestasi-prestasi mahasiswanya. Namun untuk saat ini jurusan Teknik Informatika belum memiliki sistem untuk mendokumentasikan prestasi-prestasi mahasiswanya. Untuk saat ini dalam proses pendokumentasiannya masih menggunakan *spreadsheet* pada *Google Drive* seperti yang ditunjukan pada Gambar 1.1.

(LOLOS KE BABAK KE 2, ONLINE HACKING)								
KEMAHANAN JARINGAN								
KCI	NAMA TEAM	NRP	NAMA	NRP	NAMA	NRP	NAMA	PEMBERING
1	ARDO	512100020	Rizka Filqadar	512100207	Andrya Daniel Silalahi	512100103	Dicky Inwanto	Baskoro Adi Pratomo, S.Kom, M.Kom
2	BerkumDanMeramu	512100095	Yusro Tsagora	512100071	Djuned Fernando Djusdek	512100013	Franciskus Gusli Nugrah Dwika Setiawan	Baskoro Adi Pratomo, S.Kom, M.Kom
3	Bombur	511100012	Putu Wiramaswara Widya	511100115	Ali Arief	511100063	Ihsan Hendriyasa	Baskoro Adi Pratomo, S.Kom, M.Kom
4	DEF	511100145	Enich Hendra Putra Alwando	511100130	Faldi Rianda	511100128	Danang Pionira Nugraha	Baskoro Adi Pratomo, S.Kom, M.Kom
5	dot AVI	512100094	Muhamad Ardhinata J.	512100043	M. Vijay Fathur Rahman	512100101	R. M. Iskandar Zulkarnaen	Baskoro Adi Pratomo, S.Kom, M.Kom
6	FRP	512100001	Muhammad Fakhrian Noor	512100019	Bima Nisrina Marjidi	512100190	Rahma Fida Fadhliah	Baskoro Adi Pratomo, S.Kom, M.Kom
7	IDC	513100151	Danang Adi Nugroho	513100121	Cayza Angka Maulana	513100166	Ihsan Prasetya	Baskoro Adi Pratomo, S.Kom, M.Kom

Gambar 1.1 Dokumentasi Prestasi Menggunakan Spreadsheet

Sistem dokumentasi yang masih menggunakan *Google Drive* tentunya tidak efisien jika harus memuat data prestasi yang semakin besar dari tahun ke tahun. Untuk mengetahui grafik prestasi dan mengetahui prestasi-prestasi terbaik tidak

dapat dilakukan jika bentuk dokumentasi masih dalam bentuk *file spreadsheet*. Selain berupa data hasil kemenangan, dokumentasi atau laporan keikutsertaan lomba juga diperlukan agar dapat memotivasi mahasiswa lain dalam mengikuti perlombaan. Untuk dokumentasi atau laporan perlombaan, jurusan Teknik Informatika masih menggunakan grup Facebook seperti yang ditunjukkan pada Gambar 1.2. Dimana dokumentasi dan laporan hasil perlombaan belum terekam dengan baik.



Gambar 1.2 Grup IFRITS Winning Team

Dengan adanya permasalahan tersebut, dalam Tugas Akhir ini dibuat sebuah sistem informasi berbasis website untuk dokumentasi prestasi mahasiswa. Data prestasi yang telah terdokumentasi digunakan untuk mendapatkan rekomendasi prestasi dan rekomendasi mahasiswa. Rekomendasi prestasi diperoleh dengan metode Fuzzy Multi-Attribute Decision Making Simple Additive Weighting (SAW) dan rekomendasi mahasiswa diperoleh dengan memetakan cabang kegiatan dengan rumpun mata kuliah.

Rekomendasi prestasi diharapkan mampu membantu dalam penentuan pemilihan prioritas keikutsertaan cabang kegiatan dengan mempertimbangkan peringkat rekomendasi prestasi terbaik. Sedangkan rekomendasi mahasiswa

digunakan untuk memperoleh daftar nama mahasiswa yang direkomendasikan mengikuti suatu cabang kegiatan.

1.2. Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Membuat sistem pendokumentasian prestasi mahasiswa jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya.
2. Mengimplentasikan *Fuzzy Multi-Attribute Decision Making Simple Additive Weighting* pada penentuan alternatif-alternatif prestasi terbaik jurusan teknik Informatika untuk membantu pemetaan fokus pembinaan perlombaan Institut Teknologi Sepuluh Nopember Surabaya (ITS).

1.3. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini sebagai berikut:

1. Menyediakan sistem pendokumentasian prestasi mahasiswa jurusan Teknik Informatika.
2. Menerapkan metode Fuzzy Multi-Attribute Decision Making Simple Additive Weighting untuk menentukan alternatif-alternatif prestasi terbaik jurusan Teknik Informatika untuk membantu pemetaan fokus pembinaan perlombaan.

1.4. Batasan Permasalahan

Beberapa batasan dalam Tugas Akhir ini adalah sebagai berikut:

1. Studi kasus yang dibahas adalah jurusan Teknik Informatika di perguruan tinggi Institut Teknologi Sepuluh Nopember Surabaya (ITS).
2. Data yang digunakan pada penentuan prestasi terbaik mahasiswa pada perlombaan adalah data yang

diperoleh dari data prestasi mahasiswa Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya.

3. Bahasa pemrograman yang digunakan adalah C# dengan *framework* .NET 4.5.

1.5. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini adalah:

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi mengenai hal-hal pendukung pengerjaan Tugas Akhir. Pengumpulan ini dimaksudkan untuk melakukan analisis dan perancangan sistem. Adapun literatur yang dipakai adalah:

- a. Sistem Terkait Dokumentasi Prestasi Jurusan Teknik Informatika
- b. Teori *Fuzzy Multi-Attribute Decision Making Simple Additive Weighting*
- c. Teori konsep MVC(*Model, View, Control*) Pattern
- d. Teori konsep ASP.NET
- e. Teori konsep Bootstrap
- f. Teori konsep ASP.NET MVC 4
- g. Teori Entity Framework 5
- h. Teori penggunaan EF 5 pada ASP.NET MVC 4
- i. Teori JQuery dan JSON pada ASP.NET MVC4

2. Analisis dan Perancangan Sistem

Pada tahapan ini, penulis melakukan analisis dan desain aplikasi. Analisis kebutuhan dan perancangan aplikasi, dilakukan untuk merumuskan spesifikasi kebutuhan aplikasi, mendapatkan kebutuhan-kebutuhan dalam perhitungan *Fuzzy Multi-Attribute Decision Making Simple Additive Weighting*. Adapun tahap desain dilakukan untuk memodelkan hasil analisis dalam bentuk desain arsitektur, diagram kasus penggunaan, diagram sekuensial, diagram kelas, desain

antarmuka dan desain-desain pendukung lain dalam pembangunan aplikasi.

3. Implementasi

Pada tahap ini dilakukan implementasi perangkat lunak ke dalam bentuk kode program. Adapun perincian implementasi kaskas adalah sebagai berikut:

- a. Implementasi Bootstrap pada ASP.NET
- b. Implementasi ASP.NET MVC 4
- c. Implementasi data akses Entity Framework 5 pada ASP.NET MVC 4.
- d. Implementasi *Fuzzy Multi-Attribute Decision Making Simple Additive Weighting*.

4. Pengujian dan Evaluasi

Tahapan ini digunakan untuk melakukan pengujian dan evaluasi pada aplikasi. Tahapan ini bertujuan untuk mengetahui kesalahan-kesalahan yang ada pada aplikasi sehingga dapat dilakukan perbaikan terhadap aplikasi itu sendiri. Pengujian menggunakan metode *black box* dan pengujian pengguna.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.6. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk memberikan gambaran mengenai Tugas Akhir. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan, rumusan permasalahan, batasan masalah,

metodologi yang digunakan, dan sistematika penulisan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan kakas.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak kakas dan implementasi fitur-fitur penunjang kakas.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode pengujian subjektif untuk mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak dan pengujian fungsionalitas kakas.

Bab VI Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

BAB II

DASAR TEORI

Pada bab ini membahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir. Adapun teori-teori tersebut meliputi Fuzzy Multi-Attribute Decision Making Simple Additive Weighting, MVC (Model, View, Controller) Pattern, ASP.NET, Bootstrap, ASP.NET MVC 4, Entity Framework 5, JQuery, Json dan lainnya.

2.1. Sistem Dokumentasi Prestasi Jurusan Teknik Informatika

Saat ini jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember masih menggunakan spreadsheet pada *Google Drive* dan grup *Facebook* untuk dokumentasi prestasi. Hal ini tentunya tidak efektif jika harus memuat data prestasi yang semakin besar dari tahun ke tahun. Sehingga dibutuhkan sebuah sistem untuk mempermudah proses dokumentasi prestasi.

2.2. *Fuzzy Multi-Attribute Decision Making Simple Additive Weighting*

Fuzzy Multi-Attribute Decision Making adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. FMADM adalah sebuah inti penentuan nilai bobot untuk setiap atribut, kemudian diikuti dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan.

2.2.1. *Simple Additive Weighting (SAW)*

Metode SAW sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif dari semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang

dapat diperbandingkan dengan semua rating alternatif yang ada. Proses normalisasi ini ditunjukkan oleh persamaan 2.1[2]:

$\frac{x_{ij}}{\text{Max}_i x_{ij}}$ jika j adalah atribut keuntungan (*benefit*)

$\frac{\text{Min}_i x_{ij}}{x_{ij}}$ jika j adalah atribut biaya (*cost*):

Dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} \end{cases} \quad 2.1$$

pada atribut C_j ; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$. Nilai preferensi untuk setiap alternatif (V_i) diberikan pada persamaan 2.2:

V_i = nilai profesi

$$V_i = \sum_{j=1}^n w_j r_{ij} \quad 2.2$$

w_j = bobot rangking

r_{ij} = rating kinerja ternormalisasi

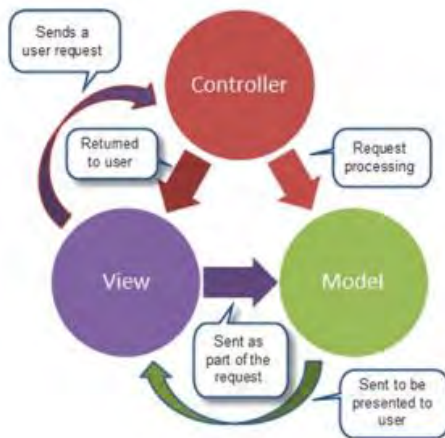
Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih. Berikut adalah langkah-langkah metode FMADM SAW [2].

1. Menentukan alternatif A_i
2. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu (C_i).
3. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
4. Membuat matriks keputusan berdasarkan kriteria (C_i).

5. Membuat normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R .
6. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi.

2.3. *MVC (Model, View, Controller) Pattern*

MVC merupakan arsitektur pembangunan sistem yang memisahkan sistem ke dalam tiga komponen utama, yaitu *model*, *view*, dan *controller*. Arsitektur Model-View-Controller ditunjukkan pada Gambar 2.1.



Gambar 2.1 Arsitektur Model-View-Controller [3]

Pada aturan MVC, data aplikasi dipanggil oleh *model* dan presentasi data dipanggil oleh *view*. *Model* dan *view* merupakan hal pasif. Masing-masing menunggu permintaan. Hal yang berfungsi untuk memanggil keduanya adalah *controller*.

Controller adalah penggerak dan penghubung pada arsitektur MVC. *Controller* menunggu permintaan pengguna. Ketika permintaan tiba, *controller* mengambil beberapa data dari *model* atau memberikan informasi kepada *model* untuk memodifikasi beberapa data. Kemudian *controller* ke *view*. *View* menampilkan data kepada pengguna sesuai dengan permintaan yang diberikan kepada sistem.

Model berfungsi untuk menyimpan model dari sistem. Untuk data permanen, biasanya disimpan dalam basis data. Komponen ini tidak bisa mengakses kelas *view* dan *controller*. Dia bersifat seperti bagian persistensi dari sistem.

View menyimpan kelas-kelas yang berhubungan dengan antarmuka. Kelas ini yang akan mengakses kelas-kelas yang ada di *model* dan *controller*. Biasanya kelas-kelas ini dibangun dengan HTML. Kelas-kelas ini digunakan untuk menampilkan data yang diambil dari model.

Controller merupakan komponen yang menyimpan kelas-kelas yang mengatur hubungan antara *model* dan *view*. Kelas ini menerima kejadian dari luar, berinteraksi dengan *model* dan menampilkan ke *view* kepada pengguna [3].

2.4. ASP.NET

ASP.NET merupakan *framework* untuk membangun web. Pada ASP.NET terdapat tiga teknologi untuk pembangunan aplikasi *web* yaitu ASP.NET Web Form, ASP.NET Web Pages, dan ASP.NET MVC. ASP.NET menjadi sebuah web platform yang menyediakan seluruh servis untuk membangun aplikasi web yang berbasis pada kelas *enterprise-server*. ASP.NET dibangun pada .NET Framework Sehingga semua fitur pada .NET Framework tersedia pada ASP.NET [3].

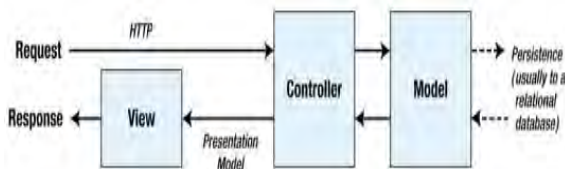
2.5. Bootstrap

Bootstrap merupakan *front-end framework* yang paling banyak digunakan saat ini. Dalam bootstrap terdapat scaffolding, HTML, CSS, komponen, plugin javascript, dan *customize*.

Bootstrap sendiri sekarang telah dikembangkan oleh banyak perusahaan penyedia *template* dan *interface*. Salah satunya adalah perusahaan BootstrapMaster. Salah satu *template* yang ditawarkan oleh perusahaan ini adalah model metro [4].

2.6. ASP.NET MVC 4

ASP.NET MVC 4 adalah kerangka kerja yang digunakan untuk pembangunan aplikasi web yang menggunakan pola *model-view-controller*. *Model* terdiri dari kelas-kelas yang menangani data dan logika bisnis. Proses penggunaan data model dilakukan melalui *controller* berdasarkan *user request*, kemudian *controller* mengirim *view* yang ditampilkan pada *browser*. Pada ASP.NET MVC, *controller* diimplementasikan dalam bentuk kelas C# yang diturunkan dari kelas ***System.Web.Mvc.Controller***. Setiap *method public* yang diturunkan dari ***Controller*** dinamakan *action method*, dimana dia berasosiasi dengan URL yang bisa dikonfigurasi melalui sistem routing ASP.NET. Ketika sebuah *request* dikirimkan pada URL yang berasosiasi dengan *action method*, perintah pada kelas *controller* akan mengeksekusi beberapa operasi pada *model* dan kemudian memilih sebuah *view* untuk ditampilkan pada client. Pada Gambar 2.2 akan ditunjukkan interaksi antara *controller*, *model*, dan *view* pada ASP.NET MVC 4.



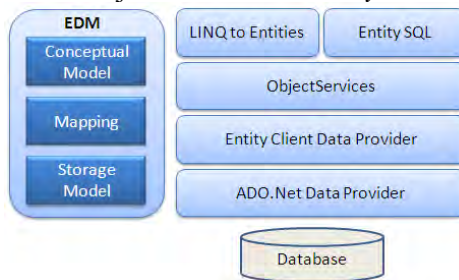
Gambar 2.2 Interaksi Aplikasi MVC [5]

Kerangka kerja ASP.NET MVC 4 menyediakan pilihan *view engines* diantaranya adalah ASPX dan RAZOR. ASP.NET MVC 4 tidak mengimplementasikan constraint pada model. Untuk membuat sebuah model dapat menggunakan objek C# dan mengimplementasikannya menggunakan database, kerangka kerja

ORM (Object Relational Mapping) atau perangkat lainnya yang mendukung .NET [5].

2.7. ADO.NET Entity Framework 5

ADO.NET Entity Framework adalah *Object Relational Mapping (ORM) framework* yang memungkinkan pengembang untuk bekerja dengan data relasional sebagai objek-domain tertentu, menghilangkan kebutuhan untuk sebagian besar akses data. Menggunakan *Entity Framework*, pengembang mengeluarkan *query* menggunakan LINQ, kemudian mengambil dan memanipulasi data sebagai objek. Implementasi ORM *Entity Framework* menyediakan layanan seperti perubahan pelacakan, resolusi identitas, *lazy loading*, dan terjemahan permintaan sehingga pengembang dapat fokus pada aplikasi-spesifik logika bisnis daripada fundamental akses data. *Object Relational Mapping (ORM) framework* merupakan peningkatan dari ADO.NET yang memberikan pengembang mekanisme otomatis untuk mengakses & menyimpan data dalam database, dan untuk bekerja dengan hasil, selain *DataReader* dan *DataSet* [6]. Pada Gambar 2.3 akan ditunjukkan arsitektur *Entity Framework*.



Gambar 2.3 Arsitektur Entity Framework [6]

Berikut akan dijelaskan setiap bagian dari arsitektur *Entity Framework*:

- a) **EDM (Entity Data Model):** EDM terdiri tiga bagian utama yaitu *conceptual model*, *mapping* dan *model storage*.

- b) **Conceptual Model:** *Conceptual Model* berisi kelas model dan hubungan mereka. Ini akan menjadi independen dari desain tabel database Anda.
- c) **Model Storage:** *Model Storage* adalah model desain database yang meliputi *tables*, *views*, *store procedure*, *relationships*, dan *keys*.
- d) **Mapping:** *Mapping* terdiri dari informasi tentang bagaimana *conceptual model* dipetakan ke *storage model*.
- e) **LINQ to Entities (L2E):** *LINQ to Entitas* adalah bahasa *query* yang digunakan untuk menuliskan *query* terhadap model objek. Ia mengembalikan entitas, yang didefinisikan dalam *conceptual model*.
- f) **Entity SQL:** *Entity SQL* adalah bahasa *query* lain seperti LINQ to Entity.
- g) **Object Service:** *Object Service* adalah titik masuk utama untuk mengakses data dari database dan untuk mengembalikannya. *Object Service* bertanggung jawab untuk perwujudan, yang merupakan proses mengubah data yang kembali dari *entity client data provider* (lapisan berikutnya) ke *entity object structure*.
- h) **Entity Client Data Provider:** Tanggung jawab utama lapisan ini adalah untuk mengkonversi L2E atau *query* Entity SQL ke dalam *query* SQL yang dipahami oleh database. Ini adalah bagian dari komunikasi antara penyedia data ADO.Net yang pada gilirannya mengirim atau mengambil data dari database.
- i) **ADO.Net Data Provider:** Lapisan ini berkomunikasi dengan database menggunakan ADO.Net standar [7].

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tahapan analisis dan perancangan Tugas Akhir. Analisis meliputi kebutuhan-kebutuhan yang diperlukan oleh perangkat lunak. Dari hasil analisis yang telah dilakukan, kemudian dilakukan tahapan perancangan sistem. Perancangan perangkat lunak direpresentasikan dengan diagram UML (*Unified Modelling Language*).

3.1. Analisis

Dalam tahapan analisis, penulis membagi ke dalam beberapa tahapan antara lain analisis permasalahan, deskripsi umum sistem, dan kebutuhan perangkat lunak.

3.1.1. Analisis Permasalahan

Teknik Informatika Institut Teknologi Sepuluh Nopember selalu berupaya untuk memfasilitasi mahasiswanya mencapai prestasi terbaik. Salah satu upaya yang dilakukan adalah melakukan dokumentasi terhadap prestasi-prestasi mahasiswanya. Dokumentasi masih menggunakan spreadsheet pada Google Drive dan grup Facebook, hal ini tidak efektif jika harus memuat data prestasi yang semakin besar dari tahun ke tahun.

Dengan adanya permasalahan tersebut, dalam Tugas Akhir ini dibuat sebuah sistem informasi berbasis website untuk dokumentasi prestasi mahasiswa. Data prestasi yang telah terdokumentasi digunakan untuk mendapatkan rekomendasi prestasi dan rekomendasi mahasiswa. Rekomendasi prestasi diperoleh dengan metode *Fuzzy Multi-Attribute Decision Making Simple Additive Weighting* (SAW) dan rekomendasi mahasiswa diperoleh dengan memetakan cabang kegiatan dengan rumpun mata kuliah.

Rekomendasi prestasi diharapkan mampu membantu dalam penentuan pemilihan prioritas keikutsertaan cabang

kegiatan dengan mempertimbangkan peringkat rekomendasi prestasi terbaik. Sedangkan rekomendasi mahasiswa digunakan untuk memperoleh daftar nama mahasiswa yang direkomendasikan mengikuti suatu cabang kegiatan.

3.1.2. Deskripsi Umum Perangkat Lunak

Perangkat lunak yang dibangun adalah aplikasi berbasis web yang dibangun menggunakan kerangka kerja ASP.NET MVC 4 dan Entity Framework 5. Aplikasi berfungsi untuk dokumentasi prestasi mahasiswa Teknik Informatika Institut Teknologi Sepuluh Nopember. Dokumentasi prestasi meliputi data prestasi, Data yang telah terdokumentasi dalam sistem akan dipergunakan untuk menentukan prestasi-prestasi terbaik berdasarkan alternatif prestasi-prestasi terbaik.

3.1.3. Spesifikasi Kebutuhan Perangkat Lunak

Dalam subbab ini membahas spesifikasi kebutuhan fungsional sistem dari hasil analisis penulis. Bagian ini berisi semua kebutuhan perangkat lunak yang ditulis dalam bentuk kebutuhan fungsional, diagram kasus penggunaan, skenario kasus penggunaan, diagram aktivitas, dan diagram sekuens.

3.1.3.1. Kebutuhan Fungsional

Kebutuhan fungsional merupakan proses-proses yang harus ada pada sistem sehingga sistem dapat berjalan dengan baik, fungsi yang merupakan kebutuhan utama dari sistem. Kebutuhan fungsional mendefinisikan layanan yang harus dimiliki oleh sistem, bagaimana reaksi sistem terhadap masukan yang ada dan apa yang dilakukan sistem pada situasi khusus. Adapun kebutuhan fungsional sistem juara.if.its.ac.id ini dijelaskan pada Tabel 3.1.

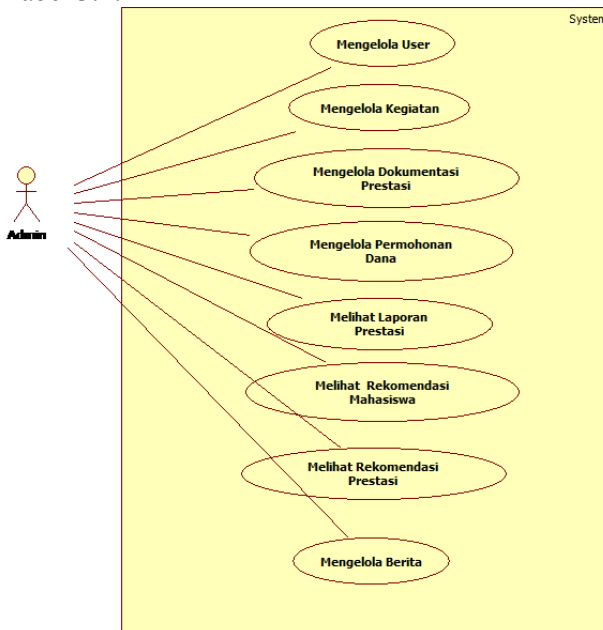
Tabel 3.1 Daftar Kode Kebutuhan Fungsional Sistem

Kode Kebutuhan Fungsional	Kebutuhan Fungsional	Deskripsi
FR01	Dokumentasi Prestasi	Fungsi ini digunakan untuk melakukan dokumentasi data prestasi mahasiswa.
FR02	Permohonan Dana	Fungsi ini untuk menangani proses transaksi permohonan dana bagi mahasiswa yang prestasinya telah berstatus finalis.
FR03	Manajemen Rekomendasi Mahasiswa	Fungsi ini untuk melakukan dokumentasi nilai mahasiswa dan melakukan perhitungan nilai mahasiswa berdasarkan pemetaan RMK dan cabang kegiatan untuk diperoleh rekomendasi nama-nama mahasiswa untuk setiap cabang kegiatan
FR04	Manajemen Proses SAW (Simple Additive Weighting)	Fungsi ini digunakan untuk melakukan perhitungan proses SAW dan menampilkan rekomendasi cabang kegiatan yang mempunyai prestasi dengan bobot terbaik.

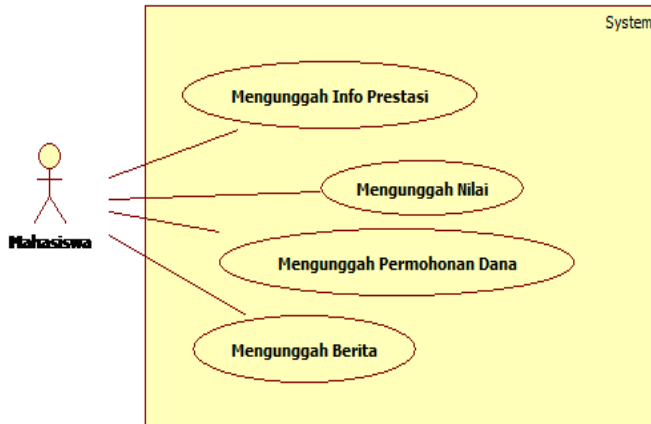
Kode Kebutuhan Fungsional	Kebutuhan Fungsional	Deskripsi
FR05	Manajemen Berita	Fungsi ini digunakan untuk manajemen berita yang submit oleh mahasiswa dan admin.

3.1.3.2. Diagram Kasus Penggunaan

Kasus penggunaan sistem berdasarkan hasil analisis kebutuhan fungsional sistem dijelaskan secara rinci pada subbab ini. Ada 12 kasus penggunaan pada sistem ini sebagaimana yang ada pada Gambar 3.1 dan Gambar 3.2. Penjelasan kasus penggunaan sistem lebih rinci dapat dilihat pada Tabel 3.2.



Gambar 3.1 Diagram Kasus Penggunaan untuk Aktor Admin



Gambar 3.2 Diagram Kasus Penggunaan untuk Aktor Mahasiswa

Tabel 3.2 Daftar Kode Diagram Kasus Penggunaan

Kode Kasus Penggunaan	Nama
UC01	Mengelola User
UC02	Mengelola Kegiatan
UC03	Mengelola Dokumentasi Prestasi
UC04	Mengelola Permohonan Dana
UC05	Melihat Laporan Prestasi
UC06	Melihat Rekomendasi Mahasiswa
UC07	Melihat Rekomendasi Prestasi
UC08	Mengelola Berita
UC09	Mengunggah Prestasi
UC10	Mengunggah Nilai
UC11	Mengunggah PerMohonan Dana
UC12	Mengunggah Berita

3.1.3.3. Skenario Kasus Penggunaan

1. Mengelola Data User

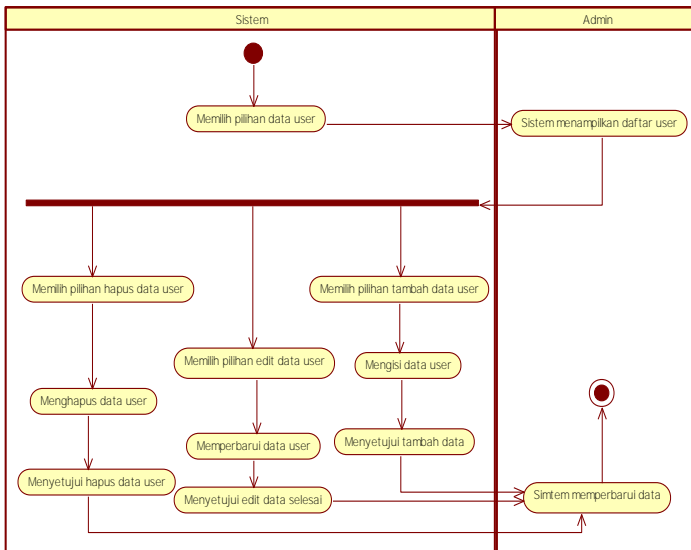
Pada kasus penggunaan mengelola data *user*, sistem menerima masukan dari pengguna berupa data *user*. Hasil akhir dari kasus kegunaan ini, sistem akan menyimpan data *user*

dalam database dan admin dapat mengelola data dengan menambah, memperbarui, atau menghapus data *user*. Data *user* ini terbagi menjadi tiga yaitu data pengguna, data mahasiswa, dan data dosen. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 3.3.

Tabel 3.3 Skenario Mengelola Data User

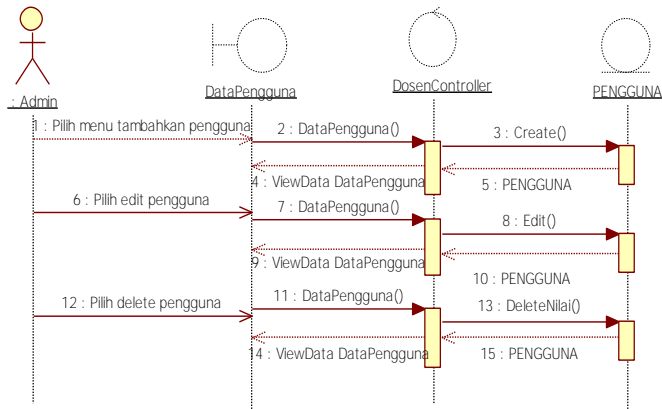
Nama Kasus Penggunaan	Mengelola Data <i>User</i>
Nomor	UC01
Deskripsi	Admin menambahkan data <i>user</i> . Selain itu admin juga dapat mengubah atau menghapus data <i>user</i> .
Aktor	Admin
Kondisi Awal	Admin telah masuk dalam halaman <i>dashboard</i> admin.
Alur Normal	1. Admin memilih pilihan data <i>user</i>
	2. Sistem menampilkan daftar data <i>user</i>
	3. Admin memilih pilihan <i>edit</i> data user A1. Admin memilih pilihan tambah data <i>user</i> A2. Admin memilih pilihan hapus data <i>user</i>
	4. Admin memperbarui data <i>user</i>
	5. Admin menyetujui <i>edit</i> data <i>user</i> selesai
Alur Alternatif	A1. Admin memilih pilihan tambah data <i>user</i> A1.1 Admin mengisi data <i>user</i> A1.2 Admin menyetujui tambahkan data A2. Admin memilih pilihan hapus data <i>user</i> A2.1 Admin memilih hapus data <i>user</i> A2.2 Admin menyetujui hapus data <i>user</i>
Kondisi Akhir	Data <i>user</i> berhasil ditambahkan dalam <i>database</i> atau berhasil diubah.

Berdasarkan skenario kasus penggunaan untuk mengelola data *user* yang ada pada tabel Tabel 3.3. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola data *user* dapat dilihat pada Gambar 3.3. Dari diagram aktivitas yang ada pada Gambar 3.3 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 3.4.



Gambar 3.3 Diagram Aktivitas Kasus Penggunaan Mengelola User

Pada kasus penggunaan ini, aktor berperan adalah admin master. Aktor tersebut memiliki hak akses dalam melakukan pengelolaan informasi data *user* dalam database melalui antarmuka *dashboard* admin master. Admin master juga dapat melakukan penambahan, pengubahan, atau penghapusan data *user* dalam *database* melalui antarmuka *dashboard* admin master.



Gambar 3.4 Diagram Sekuens dari Kasus Penggunaan Mengelola *User*

2. Mengelola Dokumentasi Prestasi

Pada kasus penggunaan mengelola dokumentasi prestasi, sistem menerima masukan dari pengguna berupa data prestasi. Hasil akhir dari kasus kegunaan ini, sistem akan menyimpan data prestasi dalam database dan admin dapat mengelola data dengan menambah, memperbarui, atau menghapus data prestasi. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.1 Lampiran A.

Berdasarkan skenario kasus penggunaan untuk mengelola data prestasi yang ada pada Tabel 8.1 Lampiran A. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola data prestasi dapat dilihat pada Gambar 8.1 Lampiran A.

Dari diagram aktivitas yang ada pada Gambar 8.1 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.11 Lampiran.

Pada kasus penggunaan ini, aktor berperan adalah admin master. Aktor tersebut memiliki hak akses dalam

melakukan pengelolaan informasi data prestasi dalam database melalui antarmuka *dashboard* admin master. Admin master juga dapat melakukan penambahan, pengubahan, atau penghapusan data prestasi dalam *database* melalui antarmuka *dashboard* admin master.

3. Mengelola Data Cabang Kegiatan

Pada kasus penggunaan mengelola data cabang kegiatan, sistem menerima masukan dari pengguna berupa data cabang kegiatan. Hasil akhir dari kasus kegunaan ini, sistem akan menyimpan data cabang kegiatan dalam database dan admin dapat mengelola data dengan menambah, memperbarui, atau menghapus data cabang kegiatan. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.2 yang dilampirkan pada halaman Lampiran A.

Berdasarkan skenario kasus penggunaan untuk mengelola data cabang kegiatan yang ada pada Tabel 8.2. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola data cabang kegiatan dapat dilihat pada Gambar 8.2 yang dilampirkan pada halaman Lampiran A.

Dari diagram aktivitas yang ada pada Gambar 8.2 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.12 Lampiran A.

Pada kasus penggunaan ini, aktor berperan adalah admin master. Aktor tersebut memiliki hak akses dalam melakukan pengelolaan informasi data cabang kegiatan dalam database melalui antarmuka *dashboard* admin master. Admin master juga dapat melakukan penambahan, pengubahan, atau penghapusan data kegiatan dalam *database* melalui antarmuka *dashboard* admin master.

4. Mengelola Permohonan Dana

Pada kasus penggunaan mengelola permohonan dana, sistem menerima masukan dari pengguna berupa permohonan

dana. Hasil akhir dari kasus kegunaan ini, sistem akan menyimpan permohonan dana dalam database dan admin dapat mengelola data dengan menambah, memperbarui, atau menghapus permohonan dana. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.3 yang dilampirkan pada halaman Lampiran A.

Berdasarkan skenario kasus penggunaan untuk mengelola data kegiatan yang ada pada tabel Tabel 8.3. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola data kegiatan dapat dilihat pada Gambar 8.3 yang dilampirkan pada halaman Lampiran A.

Dari diagram aktivitas yang ada pada Gambar 8.3 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.13 Lampiran A.

Pada kasus penggunaan ini, aktor berperan adalah admin master dan admin permohonan dana. Aktor tersebut memiliki hak akses dalam melakukan pengelolaan informasi data kegiatan dalam database melalui antarmuka *dashboard* admin master untuk admin master dan *dashboard* admin permohonan untuk admin permohonan dana. Admin master juga dapat melakukan penambahan, pengubahan, atau penghapusan data kegiatan dalam *database* melalui antarmuka *dashboard* admin master.

5. Melihat Laporan Prestasi

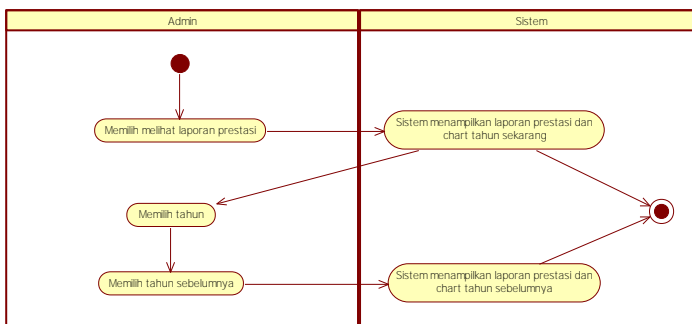
Pada kasus penggunaan melihat laporan prestasi, admin dapat melihat laporan prestasi. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 3.4 yang dilampirkan pada halaman Lampiran A.

Tabel 3.4 Skenario Melihat Laporan Prestasi

Nama Kasus Penggunaan	Melihat Laporan Prestasi
Nomor	UC05

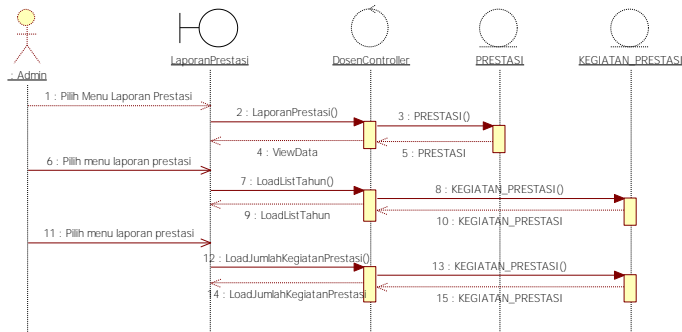
Deskripsi	Admin dapat melihat laporan prestasi dari mahasiswa yang telah memasukan data prestasi ke dalam sistem
Aktor	Admin
Kondisi Awal	Admin telah masuk dalam halaman <i>dashboard</i> admin.
Alur Normal	1. Admin memilih pilihan laporan prestasi 2. Sistem menampilkan laporan prestasi dalam bentuk <i>pie chart</i> dan daftar
	A1. Admin memilih tahun
Alur Alternatif	A1. Admin memilih tahun A1.1 Admin memilih tahun sebelumnya
Kondisi Akhir	Sistem menampilkan laporan prestasi
Lainnya	Aktor pada skenario kasus penggunaan ini adalah Admin yang mempunyai role Admin Master.

Berdasarkan skenario kasus penggunaan untuk melihat laporan prestasi yang ada pada tabel Tabel 3.4. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola data kegiatan dapat dilihat pada yang dilampirkan pada halaman Lampiran A.



Gambar 3.5 Diagram Aktivitas Melihat Laporan Prestasi

Dari diagram aktivitas yang ada pada Gambar 3.5 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram Sekuens Melihat Laporan Prestasi

Pada kasus penggunaan ini, aktor berperan adalah admin master. Aktor tersebut memiliki hak akses dalam melihat laporan prestasi melalui antarmuka *dashboard* admin master.

6. Melihat Rekomendasi Mahasiswa

Pada kasus penggunaan melihat rekomendasi mahasiswa, admin yang dapat melihat laporan prestasi. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.4 yang dilampirkan pada halaman Lampiran A.

Berdasarkan skenario kasus penggunaan untuk melihat rekomendasi mahasiswa yang ada pada tabel Tabel 8.4. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola data kegiatan dapat dilihat pada Gambar 8.4 yang dilampirkan pada halaman Lampiran A.

Dari diagram aktivitas yang ada pada Gambar 8.3 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.14 Lampiran A.

Pada kasus penggunaan ini, aktor berperan adalah admin master. Aktor tersebut memiliki hak akses dalam melihat rekomendasi mahasiswa melalui antarmuka *dashboard* admin master.

7. Melihat Rekomendasi Prestasi

Pada kasus penggunaan melihat rekomendasi prestasi, admin yang dapat melihat rekomendasi prestasi. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.5 yang dilampirkan pada halaman Lampiran A.

Berdasarkan skenario kasus penggunaan untuk melihat rekomendasi prestasi yang ada pada Tabel 8.5. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola data kegiatan dapat dilihat pada Gambar 8.5 yang dilampirkan pada halaman Lampiran A.

Dari diagram aktivitas yang ada pada Gambar 8.3 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.15 Lampiran A.

Pada kasus penggunaan ini, aktor berperan adalah admin master. Aktor tersebut memiliki hak akses dalam melihat rekomendasi prestasi melalui antarmuka *dashboard* admin master.

9. Mengelola Berita

Pada kasus penggunaan mengelola berita, sistem menerima masukan dari pengguna berupa berita. Hasil akhir dari kasus penggunaan ini, sistem akan menyimpan berita dalam database dan admin dapat mengelola data dengan menambah, memperbarui, atau menghapus berita. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.6 yang dilampirkan pada halaman Lampiran A.

Berdasarkan skenario kasus penggunaan untuk mengelola berita yang ada pada Tabel 8.6. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk

diagram aktivitas dari kasus penggunaan mengelola data kegiatan dapat dilihat pada Gambar 8.6 yang dilampirkan pada halaman Lampiran A.

Dari diagram aktivitas yang ada pada Gambar 8.3 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.16 Lampiran A.

Pada kasus penggunaan ini, aktor berperan adalah admin master dan admin berita. Aktor tersebut memiliki hak akses dalam melakukan pengelolaan berita dalam database melalui antarmuka *dashboard* admin master untuk admin master dan *dashboard* admin berita untuk admin berita. Admin master dapat melakukan penambahan, pengubahan, atau penghapusan berita dalam *database* melalui antarmuka *dashboard* admin master dan admin berita.

8. Mengunggah Info Prestasi

Pada kasus penggunaan mengunggah info prestasi, sistem menerima masukan dari pengguna berupa info prestasi. Hasil akhir dari kasus penggunaan ini, sistem akan menyimpan info prestasi dalam database dan mahasiswa dapat mengelola data dengan menambah, memperbarui, atau menghapus info prestasinya. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.7 yang dilampirkan pada halaman Lampiran A.

Berdasarkan skenario kasus penggunaan untuk mengunggah info prestasi yang ada pada Tabel 8.7. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola data kegiatan dapat dilihat pada Gambar 8.7 yang dilampirkan pada halaman Lampiran A.

Dari diagram aktivitas yang ada pada Gambar 8.3 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.17 Lampiran A.

Pada kasus penggunaan ini, aktor berperan adalah mahasiswa. Aktor tersebut memiliki hak akses dalam

mengunggah info prestasi dalam database melalui antarmuka *dashboard* mahasiswa. Mahasiswa juga dapat melakukan penambahan, pengubahan, atau penghapusan info prestasi dalam *database* melalui antarmuka *dashboard* info prestasi.

10. Mengunggah Nilai

Pada kasus penggunaan mengunggah nilai, sistem menerima masukan dari pengguna berupa nilai mahasiswa. Hasil akhir dari kasus kegunaan ini, sistem akan menyimpan nilai mahasiswa dalam database dan mahasiswa dapat mengelola data dengan menambah, memperbarui, atau menghapus nilai. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.8 yang dilampirkan pada halaman Lampiran A.

Berdasarkan skenario kasus penggunaan untuk mengelola data kegiatan yang ada pada Tabel 8.8. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola data kegiatan dapat dilihat pada Gambar 8.8.

Dari diagram aktivitas yang ada pada Gambar 8.3 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.18 Lampiran A.

Pada kasus penggunaan ini, aktor berperan adalah mahasiswa. Aktor tersebut memiliki hak akses dalam melakukan pengelolaan nilainya dalam database melalui antarmuka *dashboard* mahasiswa. Mahasiswa juga dapat melakukan penambahan, pengubahan, atau penghapusan data kegiatan dalam *database* melalui antarmuka *dashboard* mahasiswa.

11. Mengunggah Permohonan Dana

Pada kasus penggunaan mengunggah permohonan dana, sistem menerima masukan dari pengguna berupa permohonan dana. Hasil akhir dari kasus kegunaan ini, sistem akan menyimpan permohonan dana dalam database dan

mahasiswa dapat mengelola data dengan menambah, memperbarui, atau menghapus permohonan dana. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.9 yang dilampirkan pada halaman Lampiran A.

Berdasarkan skenario kasus penggunaan untuk mengelola permohonan dana yang ada pada Tabel 8.9. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengelola permohonan dana dapat dilihat pada Gambar 8.9 yang dilampirkan pada halaman Lampiran A.

Dari diagram aktivitas yang ada pada Gambar 8.3 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.19 Lampiran A.

Pada kasus penggunaan ini, aktor berperan adalah mahasiswa. Aktor tersebut memiliki hak akses dalam melakukan pengelolaan permohonan dana miliknya jika prestasi yang diajukan dalam permohonan dana telah berstatus finalis.. Mahasiswa juga dapat melakukan penambahan, pengubahan, atau penghapusan permohonan dana dalam *database* melalui antarmuka *dashboard* mahasiswa.

12. Mengunggah Berita

Pada kasus penggunaan mengunggah berita, sistem menerima masukan dari pengguna berupa berita. Hasil akhir dari kasus penggunaan ini, sistem akan menyimpan berita dalam database dan mahasiswa dapat mengelola berita dengan menambah, memperbarui, atau menghapus berita. Untuk penjelasan skenario yang lebih rinci dapat dilihat di Tabel 8.10 yang dilampirkan pada halaman Lampiran A.

Berdasarkan skenario kasus penggunaan untuk mengunggah berita yang ada pada tabel Tabel 8.10. Selanjutnya skenario tersebut digambarkan ke dalam diagram aktivitas. Untuk diagram aktivitas dari kasus penggunaan mengunggah berita dapat dilihat pada Gambar 8.10 yang dilampirkan pada halaman Lampiran A.

Dari diagram aktivitas yang ada pada Gambar 8.3 selanjutnya dibentuk diagram sekuens yang dapat dilihat pada Gambar 8.20 Lampiran A.

Pada kasus penggunaan ini, aktor berperan adalah mahasiswa. Aktor tersebut memiliki hak akses dalam mengunggah berita dalam database melalui antarmuka *dashboard* mahasiswa. Mahasiswa dapat melakukan penambahan, pengubahan, atau penghapusan berita dalam *database* melalui antarmuka *dashboard* mahasiswa. Setiap berita yang diunggah oleh mahasiswa akan tersimpan dalam bentuk *draft*. Sehingga berita tidak akan terpublikasi ke halaman berita. Berita yang telah diunggah oleh mahasiswa akan terpublikasi jika status beritanya sudah final. Pergantian status berita menjadi final dilakukan oleh admin berita atau admin master.

3.2. Perancangan Sistem

Pada subbab ini dijelaskan mengenai tahapan perancangan sistem. Perancangan sistem ini dibagi menjadi empat bagian, yaitu perancangan database, perancangan proses penentuan alternatif prestasi terbaik, perancangan arsitektur perangkat lunak, dan perancangan antarmuka pengguna.

3.2.1. Perancangan Database

Perancangan database merupakan perancangan tabel-tabel yang dibutuhkan dalam proses pembangunan sistem beserta penggambaran hubungan antar tabel. Perancangan database ini akan ditunjukkan pada gambar Gambar 8.21 dan Gambar 8.22 yang dilampirkan pada halaman Lampiran A.

3.2.1.1. Rancangan Tabel Anggota

Tabel ini digunakan untuk menyimpan data anggota dari sebuah tim prestasi. Penjelasan mengenai tabel anggota dapat dilihat pada Tabel 3.5.

Tabel 3.5 Penjelasan Tabel Anggota

Nama Kolom	Keterangan
NRP_MAHASISWA	Merupakan <i>foreign key</i> dari tabel mahasiswa
ID_PRESTASI	Merupakan <i>foreign key</i> dari tabel prestasi

3.2.1.2. Rancangan Tabel Berita

Tabel ini digunakan untuk menyimpan berita. Penjelasan mengenai tabel berita dapat dilihat pada Tabel 3.6.

Tabel 3.6 Penjelasan Tabel Berita

Nama Kolom	Keterangan
ID_BERITA	Merupakan <i>primary key</i>
ID_STATUS_BERITA	Merupakan <i>foreign key</i> dari tabel status berita
ID_PENGGUNA	Merupakan <i>foreign key</i> dari tabel pengguna
ISI_BERITA	Isi dari berita
TGL_BERITA	Tanggal berita dipublikasikan
JUDUL_BERITA	Judul dari berita
GAMBAR_BERITA	Gambar dari berita

3.2.1.3. Rancangan Tabel Cabang Kegiatan

Tabel ini digunakan untuk menyimpan cabang kegiatan prestasi. Penjelasan mengenai tabel cabang kegiatan prestasi dapat dilihat pada Tabel 3.7.

Tabel 3.7 Penjelasan Tabel Cabang Kegiatan

Nama Kolom	Keterangan
ID_CABANG	Merupakan <i>primary key</i>
ID_KEGIATAN_PRESTASI	Merupakan <i>foreign Key</i> dari tabel kegiatan prestasi
NAMA_CABANG	Merupakan nama dari cabang kegiatan

3.2.1.4. Rancangan Tabel Dosen Pembimbing

Tabel ini digunakan untuk menyimpan data dosen pembimbing. Penjelasan mengenai tabel dosbing dapat dilihat pada Tabel 3.8.

Tabel 3.8 Penjelasan Tabel Dosbing

Nama Kolom	Keterangan
ID_PRESTASI	Merupakan <i>foreign key</i> dari tabel prestasi
ID_DOSEN	Merupakan <i>foreign key</i> dari tabel dosen
NAMA_DOSEN	Nama dosen pembimbing

3.2.1.5. Rancangan Tabel Dosen

Tabel ini digunakan untuk menyimpan data dosen. Penjelasan mengenai tabel dosen dapat dilihat pada Tabel 3.9.

Tabel 3.9 Penjelasan Tabel Dosen

Nama Kolom	Keterangan
ID_DOSEN	Merupakan <i>primary key</i>
NIP_DOSEN	Merupakan <i>foreign key</i> dari tabel status berita
NAMA_DOSEN	Merupakan <i>foreign key</i> dari tabel pengguna

3.2.1.6. Rancangan Tabel *History* Prestasi

Tabel ini digunakan untuk menyimpan data history prestasi. Penjelasan mengenai tabel history prestasi dapat dilihat pada Tabel 3.10.

Tabel 3.10 Penjelasan Tabel *History* Prestasi

Nama Kolom	Keterangan
ID_HISTORY	Merupakan <i>primary key</i>
ID_STATUS_PRESTASI	Merupakan <i>foreign key</i> dari tabel status prestasi
ID_PRESTASI	Merupakan <i>foreign key</i> dari tabel prestasi
TGL_HISTORY	Tanggal dari <i>history</i> prestasi

3.2.1.7. Rancangan Tabel Jenis Kegiatan

Tabel ini digunakan untuk menyimpan jenis kegiatan. Penjelasan mengenai tabel jenis kegiatan dapat dilihat pada Tabel 3.11.

Tabel 3.11 Penjelasan Tabel Jenis Kegiatan

Nama Kolom	Keterangan
ID_JENIS_KEGIATAN	Merupakan <i>primary key</i>
NAMA_JENIS_KEGIATAN	Nama dari jenis kegiatan

3.2.1.8. Rancangan Tabel Jenis Nilai

Tabel ini digunakan untuk menyimpan data jenis nilai mahasiswa. Penjelasan mengenai tabel jenis nilai dapat dilihat pada Tabel 3.12.

Tabel 3.12 Penjelasan Tabel Jenis Nilai

Nama Kolom	Keterangan
ID_JENIS_NILAI	Merupakan <i>primary key</i>
NAMA_JENIS_NILAI	Nama jenis nilai mahasiswa
POIN_NILAI	Poin dari jenis nilai mahasiswa

3.2.1.9. Rancangan Tabel Kegiatan Prestasi

Tabel ini digunakan untuk menyimpan kegiatan prestasi. Penjelasan mengenai tabel kegiatan prestasi dapat dilihat pada Tabel 3.13.

Tabel 3.13 Penjelasan Tabel Kegiatan Prestasi

Nama Kolom	Keterangan
ID_KEGIATAN_PRESTASI	Merupakan <i>primary key</i>
ID_TINGKAT_KEGIATAN	Merupakan <i>foreign Key</i> dari tabel tingkat kegiatan
ID_JENIS_KEGIATAN	Merupakan <i>foreign Key</i> dari tabel jenis kegiatan
NAMA_KEGIATAN	Nama dari kegiatan prestasi
TAHUN_KEGIATAN	Nama dari tahun kegiatan

3.2.1.10. Rancangan Tabel Kriteria

Tabel ini digunakan untuk menyimpan kriteria dari proses *Simple Additive Weighting* untuk menentukan rekomendasi prestasi dengan poin terbaik. Penjelasan mengenai tabel kriteria dapat dilihat pada Tabel 3.14.

Tabel 3.14 Penjelasan Tabel Kriteria

Nama Kolom	Keterangan
ID_JENIS_NILAI	Merupakan <i>primary key</i>
NAMA_JENIS_NILAI	Nama jenis nilai mahasiswa
POIN_NILAI	Poin dari jenis nilai mahasiswa

3.2.1.11. Rancangan Tabel Mahasiswa

Tabel ini digunakan untuk menyimpan data mahasiswa. Penjelasan mengenai tabel mahasiswa dapat dilihat pada Tabel 3.15.

Tabel 3.15 Penjelasan Tabel Mahasiswa

Nama Kolom	Keterangan
NRP_MAHASISWA	Merupakan <i>primary key</i>
ID_STATUS_MHS	Merupakan <i>foreign Key</i> dari tabel status mahasiswa
NAMA_MAHASISWA	Nama dari mahasiswa

3.2.1.12. Rancangan Tabel Mata Kuliah

Tabel ini digunakan untuk menyimpan data mata kuliah teknik informatika. Penjelasan mengenai tabel mata kuliah dapat dilihat pada Tabel 3.16.

Tabel 3.16 Penjelasan Tabel Mata Kuliah

Nama Kolom	Keterangan
ID_MATA_KULIAH	Merupakan <i>primary key</i>
ID_RMK	Merupakan <i>foreign key</i> dari tabel RMK
NAMA_MATAKULIAH	Nama dari mata kuliah

3.2.1.13. Rancangan Tabel Nilai Mahasiswa

Tabel ini digunakan untuk menyimpan data nilai kuliah mahasiswa teknik informatika. Penjelasan mengenai tabel nilai mahasiswa dapat dilihat pada Tabel 3.17.

Tabel 3.17 Penjelasan Tabel Nilai Mahasiswa

Nama Kolom	Keterangan
ID_NILAI_MAHASISWA	Merupakan <i>primary key</i>
ID_MATA_KULIAH	Merupakan <i>foreign Key</i> dari tabel mata kuliah
ID_JENIS_NILAI	Merupakan <i>foreign Key</i> dari tabel jenis nilai
NRP_MAHASISWA	Merupakan <i>foreign Key</i> dari tabel mahasiswa

3.2.1.14. Rancangan Tabel Nilai Range

Tabel ini digunakan untuk menyimpan nilai range untuk perhitungan *Simple Additive Weighting* yang menentukan rekomendasi prestasi dengan poin terbaik. Penjelasan mengenai tabel nilai range dapat dilihat pada Tabel 3.18.

Tabel 3.18 Penjelasan Tabel Nilai Range

Nama Kolom	Keterangan
ID_NILAI_RANGE	Merupakan <i>primary key</i>
ID_VARIABEL	Merupakan <i>foreign key</i> dari tabel variabel
ID_KRITERIA	Merupakan <i>foreign key</i> dari tabel kriteria
NILAI_MULAI	Nilai mulai untuk <i>range</i>
NILAI_BATAS	Nilai batas untuk <i>range</i>

3.2.1.15. Rancangan Tabel Pengguna

Tabel ini digunakan untuk menyimpan data pengguna yang digunakan untuk login pada website JUARA. Penjelasan mengenai tabel pengguna dapat dilihat pada Tabel 3.19.

Tabel 3.19 Penjelasan Tabel Pengguna

Nama Kolom	Keterangan
ID_PENGGUNA	Merupakan <i>primary key</i>
ID_ROLE	Merupakan <i>foreign key</i> dari tabel <i>role</i>
USERNAME	Username untuk pengguna
PASSWORD	Password untuk pengguna

3.2.1.16. Rancangan Tabel Peringkat Juara

Tabel ini digunakan untuk menyimpan peringkat kegiatan prestasi. Penjelasan mengenai tabel peringkat kegiatan prestasi dapat dilihat pada Tabel 3.20.

Tabel 3.20 Penjelasan Tabel Peringkat Kegiatan

Nama Kolom	Keterangan
ID_PERINGKAT_JUARA	Merupakan <i>primary key</i>
LEVEL_JUARA	Merupakan level juara untuk kegiatan prestasi
POIN_JUARA	Merupakan poin juara untuk kegiatan prestasi

3.2.1.17. Rancangan Tabel Permohonan Dana

Tabel ini digunakan untuk menyimpan data permohonan dana. Penjelasan mengenai tabel permohonan dana dapat dilihat pada Tabel 3.21.

Tabel 3.21 Penjelasan Tabel Permohonan Dana

Nama Kolom	Keterangan
ID_PERMOHONAN	Merupakan <i>primary key</i>
ID_STATUS_PERMOHONAN	Merupakan <i>foreign key</i> dari tabel status permohonan
ID_PRESTASI	Merupakan <i>foreign key</i> dari tabel prestasi
JUDUL_PROPOSAL	Judul dari proposal permohonan dana
FILE_PROPOSAL	File dari proposal permohonan dana

3.2.1.18. Rancangan Tabel Prestasi

Tabel ini digunakan untuk menyimpan data prestasi mahasiswa. Penjelasan mengenai tabel prestasi dapat dilihat pada Tabel 3.22.

Tabel 3.22 Penjelasan Tabel Prestasi

Nama Kolom	Keterangan
ID_PRESTASI	Merupakan <i>primary key</i>
ID_PENGGUNA	Merupakan <i>foreign key</i> dari tabel pengguna
ID_PERINGKAT_JUARA	Merupakan <i>foreign key</i> dari tabel peringkat juara
ID_CABANG	Merupakan <i>foreign key</i> dari tabel cabang kegiatan
NAMA_TIM	Nama tim dari prestasi
JUMLAH_ANGGOTA	Jumlah anggota dari sebuah tim

3.2.1.19. Rancangan Tabel RMK

Tabel ini digunakan untuk menyimpan data rumpun mata kuliah teknik informatika. Penjelasan mengenai tabel rmk dapat dilihat pada Tabel 3.23.

Tabel 3.23 Penjelasan Tabel RMK

Nama Kolom	Keterangan
ID_RMK	Merupakan <i>primary key</i>
NAMA_RMK	Nama dari rumpun mata kuliah

3.2.1.20. Rancangan Tabel RMK Lomba

Tabel ini digunakan untuk menyimpan pemetaan antara rumpun mata kuliah dan lomba. Pemetaan ini digunakan untuk menentukan rekomendasi mahasiswa yang mempunyai nilai tertinggi dalam satu rumpun mata kuliah berdasarkan cabang kegiatan. Penjelasan mengenai tabel RMK Lomba dapat dilihat pada Tabel 3.24.

Tabel 3.24 Penjelasan Tabel RMK Lomba

Nama Kolom	Keterangan
ID_RMK	Merupakan <i>foreign key</i> dari tabel RMK
ID_CABANG	Merupakan <i>foreign key</i> dari tabel cabang kegiatan

3.2.1.21. Rancangan Tabel *Role*

Tabel ini digunakan untuk menyimpan *role* yang digunakan untuk menentukan hak akses pada website JUARA. Penjelasan mengenai tabel *role* dapat dilihat pada Tabel 3.25.

Tabel 3.25 Penjelasan Tabel *role*

Nama Kolom	Keterangan
ID_ROLE	Merupakan <i>primary key</i>
NAMA_ROLE	Merupakan nama <i>role</i> dari pengguna website JUARA

3.2.1.22. Rancangan Tabel Status Berita

Tabel ini digunakan untuk menyimpan status berita. Penjelasan mengenai tabel status berita dapat dilihat pada Tabel 3.26.

Tabel 3.26 Penjelasan Tabel Status Berita

Nama Kolom	Keterangan
ID_STATUS_BERITA	Merupakan <i>primary key</i>
NAMA_STATUS_BERITA	Nama status berita

3.2.1.23. Rancangan Tabel Status Mahasiswa

Tabel ini digunakan untuk menyimpan data status mahasiswa yang dipergunakan untuk menentukan hak akses terhadap sistem. Jika mahasiswa sudah berstatus lulus, maka mahasiswa tidak dapat melakukan login ke website JUARA. Penjelasan mengenai tabel status mahasiswa dapat dilihat pada Tabel 3.27.

Tabel 3.27 Penjelasan Tabel Status Mahasiswa

Nama Kolom	Keterangan
ID_STATUS_MHS	Merupakan <i>Primary Key</i>
NAMA_STATUS_MHS	Nama dari status mahasiswa

3.2.1.24. Rancangan Tabel Status Permohonan

Tabel ini digunakan untuk menyimpan data status permohonan dana. Penjelasan mengenai tabel status permohonan dapat dilihat pada Tabel 3.28.

Tabel 3.28 Penjelasan Tabel Status Permohonan

Nama Kolom	Keterangan
ID_STATUS_PERMOHONAN	Merupakan <i>primary key</i>
NAMA_STATUS_PERMOHONAN	Nama dari status permohonan

3.2.1.25. Rancangan Tabel Status Prestasi

Tabel ini digunakan untuk menyimpan data status prestasi. Penjelasan mengenai tabel status prestasi dapat dilihat pada Tabel 3.29.

Tabel 3.29 Penjelasan Tabel Status Prestasi

Nama Kolom	Keterangan
ID_STATUS_PRESTASI	Merupakan <i>primary key</i>
NAMA_STATUS_PRESTASI	Nama status prestasi

3.2.1.26. Rancangan Tabel Tingkat Kegiatan

Tabel ini digunakan untuk menyimpan tingkat kegiatan. Penjelasan mengenai tabel tingkat kegiatan dapat dilihat pada Tabel 3.30.

Tabel 3.30 Penjelasan Tabel Tingkat Kegiatan

Nama Kolom	Keterangan
ID_TINGKAT_KEGIATAN	Merupakan <i>primary key</i>

NAMA_TINGKAT_KEGIATAN	Nama dari tingkat kegiatan
POIN_TINGKAT	Merupakan poin dari tingkat kegiatan

3.2.1.27. Rancangan Tabel Variabel

Tabel ini digunakan untuk menyimpan variabel yang digunakan untuk proses perhitungan *Simple Additive Weighting* yang menentukan rekomendasi prestasi dengan poin terbaik. Penjelasan mengenai tabel variabel dapat dilihat pada Tabel 3.31.

Tabel 3.31 Penjelasan Tabel Variabel

Nama Kolom	Keterangan
ID_VARIABEL	Merupakan <i>primary key</i>
NAMA_VARIABEL	Nama dari variabel
BOBOT_VARIABEL	Bobot dari setiap variabel

3.2.1.28. Rancangan Tabel Rekomendasi SAW

Tabel ini digunakan untuk menyimpan hasil perhitungan *Simple Additive Weighting*. Hal ini dilakukan untuk menghindari perhitungan *Simple Additive Weighting* saat setiap dipilih menu rekomendasi mahasiswa, hal ini dilakukan untuk meringankan kerja aplikasi JUARA. Penjelasan mengenai tabel variabel dapat dilihat pada Tabel 3.32.

Tabel 3.32 Penjelasan Tabel Rekomendasi SAW

Nama Kolom	Keterangan
ID_SAW	Merupakan <i>primary key</i>
TAHUN	Tahun untuk kegiatan prestasi
KEGIATAN	Nama kegiatan prestasi
CABANG	Nama cabang kegiatan prestasi
PESERTA	Jumlah peserta cabang kegiatan
FINALIS	Jumlah finalis cabang kegiatan
JUARA	Jumlah poin juara cabang kegiatan
TINGKAT	Jumlah poin tingkat cabang kegiatan

3.2.2. Perancangan Proses Aplikasi

Pada subbab ini akan dijelaskan mengenai rancangan proses pembangunan aplikasi JUARA yang meliputi perancangan penentuan rekomendasi prestasi terbaik dan rekomendasi mahasiswa.

3.2.2.1. Proses Rekomendasi Prestasi

Rekomendasi prestasi ini digunakan untuk menentukan pilihan-pilihan fokus pembinaan berdasarkan alternatif-alternatif prestasi terbaik dengan menerapkan metode *Fuzzy Multi-Attribute Decision Making Simple Additive Weighting (FMADM SAW)*. Penentuan prestasi terbaik ini ditentukan berdasarkan perlombaan yang ikuti oleh mahasiswa. Sehingga data yang digunakan haruslah sudah terdokumentasi dalam sistem JUARA. Misalkan, mahasiswa Teknik Informatika telah mengikuti perlombaan seperti pada Tabel 3.33.

Tabel 3.33 Contoh Data Prestasi Mahasiswa Teknik Informatika

Nama Kegiatan Cabang Kegiatan Jumlah Peserta	Gemastik 2013 Keamanan Jaringan 20 Tim (1 Tim = 3 Orang)	ACM ICPC 2013 Programming 20 Tim(1 Tim = 3 Orang)	MGDW 5 Pembuatan Game 10 Tim (40 Orang)
Jumlah Finalis	2 Tim (1 Tim = 3 Orang)	9 Tim(1 Tim = 3 Orang)	2 Tim (8 Orang)
Jumlah Juara	Juara 2 Juara 3	-	Juara 1 Juara Favorit
Tingkat Perlombaan	Nasional	Internasional	Nasional

Dari data pada Tabel 3.3, akan ditentukan prestasi terbaik jurusan Teknik Informatika di antara tiga pilihan yaitu

Gemastik 2013, ACM ICPC 2013, dan MGDW 5. Kemudian menerapkan *Fuzzy Multi-Attribute Decision Making Simple Additive Weighting (SAW)*, berikut akan ditunjukkan proses dari metode SAW:

1. Menentukan alternatif A_i

Alternatif ini ditentukan dari cabang kegiatan yang telah diikuti oleh mahasiswa Teknik Informatika. Contoh dari alternatif ditunjukkan pada Tabel 3.34.

Tabel 3.34 Contoh Alternatif dari Data Prestasi Teknik Informatika

Alternatif (A_i)	Nama Alternatif
A_1	Gemastik Keamanan Jaringan 2013
A_2	ACM ICPC 2013
A_3	MGDW 5

2. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu (C_i). Dalam studi kasus Tugas Akhir ini akan ditentukan kriteria-kriteria dalam penentuan alternatif-alternatif prestasi terbaik mahasiswa Teknik Informatika dalam suatu perlombaan. Kriteria-kriteria tersebut ditunjukkan pada Tabel 3.35.

Tabel 3.35 Kriteria untuk sistem JUARA

No	Kriteria	
1	Jumlah Peserta	Jumlah Poin Kejuaraan
2	Jumlah Finalis	Poin Tingkat Perlombaan

Kriteria ini ditentukan berdasarkan data yang tersedia dalam sistem. Jika ditambahkan kriteria selain kriteria dalam tabel maka sistem tidak akan melakukan proses terhadap kriteria tersebut. Sehingga kriteria ini bersifat statis. Pada kriteria di atas terdapat satu kriteria yang mempunyai level yaitu jumlah juara. Dimana setiap level akan dikalikan dengan bobot setiap kriteria. Level-level tersebut ditunjukkan pada Tabel 3.36. Pada Tabel 3.36 ditunjukkan terdapat lima jenis level dengan jumlah poin masing-masing yang dapat ditentukan oleh admin.

Tabel 3.36 Jumlah Poin Juara

Level Juara	Poin
Juara 1	20
Juara 2	15
Juara 3	10
Juara Harapan	5
Juara Favorit	10

Kriteria-kriteria yang sudah ditentukan dinamakan sebagai (C_i), seperti pada Tabel 3.37.

Tabel 3.37 Kriteria (C_i)

Kriteria	Nama Kriteria
C_1	Jumlah Peserta
C_2	Jumlah Finalis
C_3	Jumlah Kejuaraan
C_4	Tingkat Perlombaan

3. Menentukan rating kecocokan setiap alternatif pada setiap kriteria yang telah ditentukan. Dari kriteria yang sudah ditentukan, maka dibuat suatu tingkat kepentingan kriteria berdasarkan nilai bobot yang telah ditentukan

kedalam bilangan fuzzy dengan rumus $\frac{n}{n-1}$ yang ditunjukkan pada Tabel 3.38.

Tabel 3.38 Bobot Variabel

Variabel	Fuzzy
Sangat Penting	1
Tinggi	0,75
Cukup	0,5
Rendah	0,25
Sangat Rendah	0

Berdasarkan kriteria dan rating kecocokan setiap alternatif pada setiap kriteria yang telah ditentukan, selanjutnya penjabaran bobot setiap kriteria yang ditunjukkan oleh Tabel 3.39, Tabel 3.40, Tabel 3.41, Tabel 3.42. Bobot setiap kriteria bisa diubah sesuai dengan kebutuhan.

Tabel 3.39 Kriteria dan Bobot Jumlah Peserta (C_1)

Jumlah Peserta	Bobot
Peserta 0	0
Peserta 0-20	0,25
Peserta 21-40	0,5
Peserta 41-50	0,75
Peserta >50	1

Tabel 3.40 Kriteria dan Bobot Jumlah Finalis (C_2)

Jumlah Finalis	Bobot
Finalis 0	0
Finalis 0-20	0,25
Finalis 21-40	0,5
Finalis 41-50	0,75

Jumlah Finalis	Bobot
Finalis >50	1

Tabel 3.41 Kriteria dan Bobot Poin Juara (C_3)

Poin Juara	Bobot
0	0
5-20	0,25
21-35	0,5
36-45	0,75
46-60	1

Tabel 3.42 Kriteria dan Bobot Poin Tingkat Kegiatan (C_4)

Poin Tingkat Kegiatan	Bobot
10	0
20	0,25
30	0,5
40	0,75
50	1

Dari pembobotan setiap kriteria dapat dilakukan pengecekan terhadap kriteria-kriteria yang dimiliki oleh alternatif-alternatif. Hasil pengecekan yang telah dilakukan ditampilkan dalam bentuk tabel yang ditunjukkan pada Tabel 3.43.

Tabel 3.43 Hasil Pengecekan Bobot Setiap Kriteria pada Alternatif

Kriteria

Alternatif (A)	C_1	C_2	C_3	C_4
A_1	1	0,25	0,5	0,75
A_2	1	0,5	0	1
A_3	0,5	0,25	0,5	0,75

4. Membuat matriks keputusan berdasarkan kriteria (C_i). Matriks dibentuk dari hasil pengecekan bobot setiap kriteria pada alternatif-alternatif, seperti yang ditunjukkan pada Tabel 3.43.

$$X = \begin{bmatrix} 1 & 0,25 & 0,5 & 0,75 \\ 1 & 0,5 & 0 & 1 \\ 0,5 & 0,25 & 0,5 & 0,75 \end{bmatrix}$$

5. Membuat normalisasi matriks berdasarkan persamaan 2.1 yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R. Proses normalisasi menggunakan persamaan 2.1 dengan j sebagai atribut keuntungan (*benefit*). Menggunakan atribut keuntungan karena pada studi kasus Tugas Akhir ini membutuhkan alternatif-alternatif yang mendukung pemilihan alternatif-alternatif prestasi terbaik jurusan Teknik Informatika. Hasil normalisasi ditunjukkan pada Tabel 3.44.

Normalisasi matriks untuk alternatif A_1

$$r_{11} = \frac{1}{\text{Max}\{1; 1; 0,5\}} = \frac{1}{1} = 1$$

$$r_{12} = \frac{0,25}{\text{Max}\{0,25; 0,5; 0,25\}} = \frac{0,25}{0,5} = 0,5$$

$$r_{13} = \frac{0,5}{\text{Max}\{0,5; 0; 0,5\}} = \frac{0,5}{0,5} = 1$$

$$r_{14} = \frac{0,75}{\text{Max}\{0,75; 1; 0,75\}} = \frac{0,75}{1} = 0,75$$

Normalisasi matriks untuk alternatif A_2

$$r_{21} = \frac{1}{\text{Max}\{0,5; 1; 0,5\}} = \frac{1}{1} = 0,5$$

$$r_{22} = \frac{0,5}{\text{Max}\{0,25; 0,5; 0,25\}} = \frac{0,5}{0,5} = 1$$

$$r_{23} = \frac{0}{\text{Max}\{0,5; 0; 0,5\}} = \frac{0}{0,5} = 0$$

$$r_{24} = \frac{1}{\text{Max}\{0,75; 1; 0,75\}} = \frac{1}{1} = 1$$

Normalisasi matriks untuk alternatif A_3

$$r_{31} = \frac{0,5}{\text{Max}\{0,5; 1; 0,5\}} = \frac{0,5}{1} = 0,5$$

$$r_{32} = \frac{0,25}{\text{Max}\{0,25; 0,5; 0,25\}} = \frac{0,25}{0,5} = 0,5$$

$$r_{33} = \frac{0,5}{\text{Max}\{0,5; 0; 0,5\}} = \frac{0,5}{0,5} = 1$$

$$r_{34} = \frac{0,75}{\text{Max}\{0,75; 1; 0,75\}} = \frac{0,75}{1} = 0,75$$

Tabel 3.44 Kriteria dan Bobot Ternormalisasi

Alternatif (A_i)	Kriteria			
	C_1	C_2	C_3	C_4
A_1	1	0,5	1	0,75
A_2	0,5	1	0	1
A_3	0,5	0,5	1	0,75

Dari Tabel 3.44 dibuatlah matriks ternormalisasi R.

$$R = \begin{bmatrix} 1 & 0,5 & 1 & 0,75 \\ 0,5 & 1 & 0 & 1 \\ 0,5 & 0,5 & 1 & 0,75 \end{bmatrix}$$

6. Menentukan nilai profesi yang diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan bobot rangking sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi. Nilai bobot rangking yang menjadi pengukur tingkat kepentingan ditunjukkan oleh Tabel 3.38. Penentuan nilai profesi diperoleh dengan menggunakan 2.2) dan hasil perankingan ditunjukkan pada Tabel 3.45.

W = [Sangat Tinggi; Tinggi; Cukup; Rendah; Sangat Rendah]

W = [1 ; 0,75 ; 0,5 ; 0,25 ; 0]

$$\begin{aligned} V_1 &= (1)(1) + (0,75)(0,5) + (0,5)(1) + (0,25)(0,75) \\ &= 1 + 0,375 + 0,5 + 0,25 + 0,1875 \\ &= 2,3125 \end{aligned}$$

$$\begin{aligned} V_2 &= (1)(0,5) + (0,75)(1) + (0,5)(0) + (0,25)(1) \\ &= 0,5 + 0,75 + 0 + 0,25 \\ &= 1,5 \end{aligned}$$

$$\begin{aligned} V_3 &= (1)(0,5) + (0,75)(0,5) + (0,5)(1) + (0,25)(0,75) \\ &= 0,5 + 0,375 + 0,5 + 0,25 + 0,1875 \\ &= 1,8125 \end{aligned}$$

Tabel 3.45 Hasil

Ranking	Alternatif	Value
1	A_1	2,3125
2	A_2	1,5
3	A_3	1,8125

Dari hasil diketahui bahwa A_1 (Gemastik Keamanan Jaringan 2013) adalah alternatif prestasi terbaik, sehingga dapat dijadikan sebagai alternatif pertama dalam melakukan pembinaan prestasi.

3.2.2.2. Proses Rekomendasi Mahasiswa

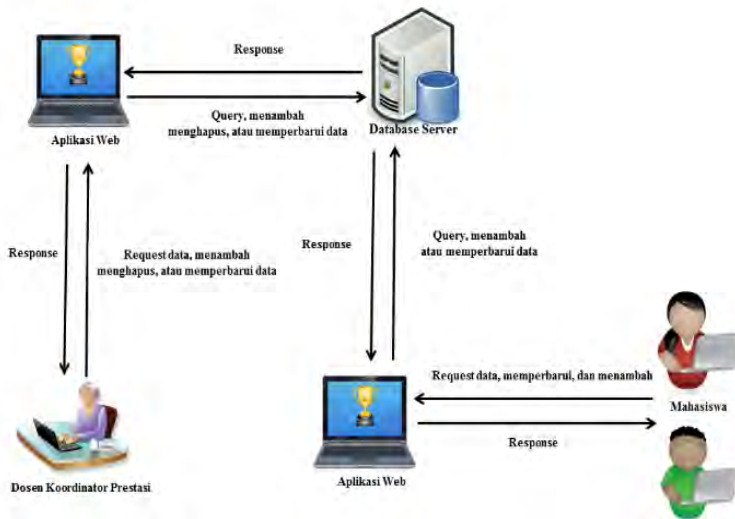
Rekomendasi nama-nama mahasiswa untuk mengikuti perlombaan berdasarkan nilai mata kuliah. Penentuan nama-nama mahasiswa yang akan diajukan untuk mengikuti perlombaan didasarkan pada pengolahan total nilai yang didapatkan mahasiswa yang dikelompokkan dalam rumpun mata kuliah. Rekomendasi mahasiswa akan dipetakan dalam cabang kegiatan berdasarkan rumpun mata kuliah.

3.2.3. Perancangan Kelas

Perancangan diagram kelas merupakan perancangan kelas-kelas yang dibutuhkan dalam proses pembangunan sistem beserta penggambaran hubungan antar kelas. Perancangan diagram kelas sistem dapat dilihat pada Gambar 8.28. Diagram kelas terbagi menjadi 2 bagian yaitu diagram kelas model dan diagram kelas kontrol. Diagram kelas model aplikasi JUARA ditunjukkan pada Gambar 8.23, sedangkan untuk kelas kontrol terbagi menjadi 4 bagian yaitu kelas kontrol HomeController pada Gambar 8.24, kelas kontrol UserController pada Gambar 8.25, kelas kontrol MahasiswaController pada Gambar 8.26, dan kelas kontrol DosenController pada Gambar 8.27.

3.2.4. Perancangan Arsitektur Sistem

Rancangan arsitektur sistem juara dapat dilihat pada Gambar 3.7.



Gambar 3.7 Rancangan Arsitektur Sistem JUARA

3.2.5. Perancangan Antarmuka Grafis

Pada subbab ini akan dijelaskan mengenai rancangan antarmuka grafis yang akan memberi fasilitas pengguna dengan sistem. Rancangan yang dibahas meliputi kontrol pada antarmuka dan ketentuan input.

3.2.5.1. Antarmuka Halaman Utama

Pada halaman utama Juara terdapat satu pilihan yaitu berita untuk melihat berita prestasi terbaru. Rancangan tampilan antarmuka halaman utama ditunjukkan pada Gambar 3.8.



Gambar 3.8 Gambar Halaman Utama Juara

3.2.5.2. Halaman *Dashboard Admin Master*

Pada halaman *dashboard* admin master terdapat enam bagian manajemen yaitu manajemen data, manajemen transaksi, manajemen rekomendasi mahasiswa, manajemen proses saw, manajemen status, dan manajemen berita. Pada setiap bagian manajemen terdapat pilihan-pilihan yang harus dikelola oleh Admin Master. Halaman Admin master ditunjukkan pada Gambar 3.10. Berikut akan dijelaskan pilihan-pilihan pada setiap bagian manajemen pada tabel Tabel 3.46. Kemudian untuk rancangan tampilan form yang digunakan untuk menambahkan data dapat dilihat pada Gambar 3.9.

Tabel 3.46 Daftar Pilihan pada Setiap Bagian Pilihan Manajemen

Manajemen Data	Data Pengguna Data Mahasiswa Data Dosen Data Prestasi Data Peringkat Kejuaraan
----------------	--

	Data Jenis Kegiatan Data Tingkat Kejuaraan Data Kegiatan Data Cabang Kegiatan
Manajemen Transaksi	Permohonan Dana Laporan Transaksi
Manajemen Rekomendasi Mahasiswa	Data Bidang Data Mata Kuliah Data Jenis Nilai Pemetaan Rekomendasi Mahasiswa
Manajemen Proses SAW	Kriteria Variabel Bobot Kriteria Rekomendasi Prestasi
Manajemen Status	Data Status Prestasi Data Status Permohonan Data Status Mahasiswa Data Status Berita Data Role
Manajemen Berita	Manajemen Berita

The image shows a login form with a light gray border. It contains two text input fields: the first is labeled 'Username' and the second is labeled 'Password'. Below the password field are two buttons: a blue button labeled 'Tambahkan' and a gray button labeled 'Batal'.

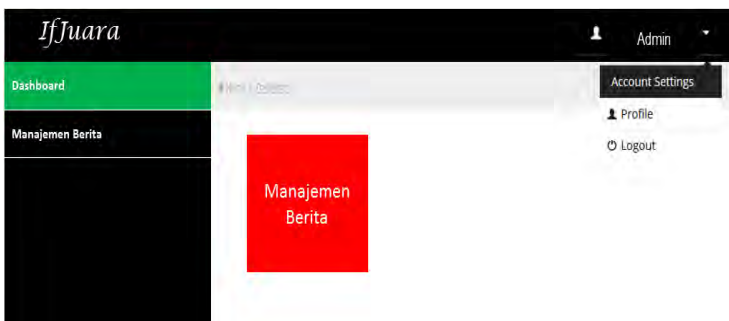
Gambar 3.9 Rancangan Form



Gambar 3.10 Rancangan Dashboard JUARA untuk Admin

3.2.5.3. Halaman *Dashboard* Admin Berita

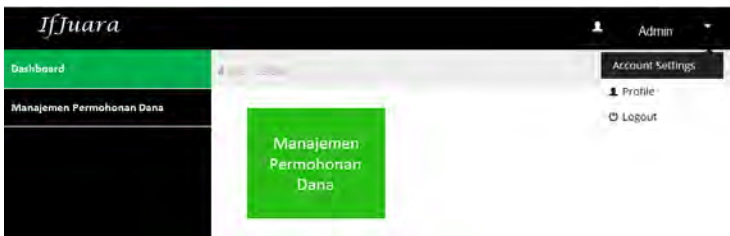
Pada halaman *dashboard* admin berita terdapat satu bagian manajemen yaitu manajemen berita. Admin berita hanya mempunyai hak akses untuk mengelola berita. Admin berita dapat mengunggah berita beserta mempunyai hak akses untuk melakukan publikasi berita yang masuk dalam daftar *draf* berita. Halaman Admin Berita ditunjukkan pada Gambar 3.11.



Gambar 3.11 Halaman *Dashboard* Admin Berita

3.2.5.4. Halaman *Dashboard* Admin Permohonan Dana

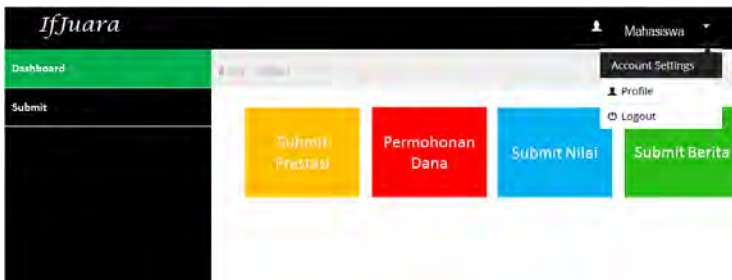
Pada halaman *dashboard* admin permohonan dana terdapat satu bagian manajemen yaitu manajemen permohonan dana. Admin permohonan hanya mempunyai hak akses untuk mengelola permohonan dana. Admin permohonan dana dapat mengubah status permohonan dana yang diajukan oleh mahasiswa yang mempunyai status prestasi finalis. Admin permohonan dana bisa mengubah statusnya menjadi revisi, ok, atau ditolak. Halaman Admin Permohonan Dana ditunjukkan pada Gambar 3.12.



Gambar 3.12 Halaman *Dashboard* Admin Permohonan Dana

3.2.5.5. Halaman *Dashboard* Mahasiswa

Pada halaman *dashboard* mahasiswa terdapat empat pilihan yaitu *submit* prestasi, permohonan dana, *submit* nilai, dan *submit* berita. Mahasiswa yang mengikuti kegiatan perlombaan atau forum wajib untuk memasukkan data prestasi melalui pilihan *submit* prestasi. Jika prestasi mahasiswa telah berstatus finalis, maka mahasiswa dapat mengajukan permohonan dana pada pilihan permohonan dana. Pada pilihan *submit* nilai, Mahasiswa disarankan untuk mengirimkan nilai mata kuliahnya, dari nilai yang telah tersimpan dalam sistem maka akan digunakan untuk menentukan rekomendasi mahasiswa. Halaman dashboard mahasiswa ditunjukkan pada Gambar 3.13.



Gambar 3.13 Halaman Dashboard Mahasiswa

BAB IV IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi pembangunan sistem berdasarkan analisis dan perancangan yang telah dijelaskan pada bab sebelumnya. Implementasi ini meliputi implementasi kelas pada lapisan data, kontrol, dan antarmuka serta implementasi tampilan antarmuka pengguna.

4.1. Lingkungan Implementasi

4.1.1. Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem adalah komputer. Spesifikasi dari perangkat tersebut adalah PC Lenovo, Intel(R) Core (TM) i3-3240 CPU @3.40GHz 3.39GHz, RAM 4.00 GB.

4.1.2. Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

1. Microsoft Windows 8.1 sebagai sistem operasi.
2. Microsoft Visual Studio 2012 sebagai tools untuk mengimplementasikan aplikasi.
3. SQL SERVER 2012 sebagai server basis data.
4. Power Designer 16.5 untuk merancang basis data.
5. Internet Information Services (IIS) sebagai *web server*.

4.2. Implementasi Proses Aplikasi

4.2.1. Implementasi Model

Lapisan data merupakan lapisan yang terdiri dari kelas-kelas yang bersifat sebagai penyimpanan. Pada lapisan ini terdiri dari atribut-atribut yang digunakan untuk menyimpan elemen-elemen pada saat membangun sistem. Lapisan ini hanya dapat diakses oleh lapisan kontrol. Lapisan merupakan

generate dari model yang dibuat dari ADO.NET Entity Framework 5. Dalam aplikasi JUARA ini terdapat 24 Kelas Model yang ditampilkan pada Tabel 4.1.

Tabel 4.1 Daftar Kelas Model

No	Nama Kelas Model
1.	STATUS_MAHASISWA
2.	MAHASISWA
3.	JENIS_KEGIATAN
4.	TINGKAT_KEGIATAN
5.	KEGIATAN_PRESTASI
6.	CABANG_KEGIATAN
7.	PERINGKAT_KEGIATAN
8.	ROLE
9.	PENGUNA
10.	PRESTASI
11.	STATUS_BERITA
12.	BERITA
13.	DOSEN
14.	STATUS_PRESTASI
15.	HISTORY_PRESTASI
16.	JENIS_NILAI
17.	KRITERIA
18.	RMK
19.	MATA_KULIAH
20.	NILAI_MAHASISWA
21.	VARIABEL
22.	NILAI_RANGE
23.	STATUS_PERMOHONAN
24.	PERMOHONAN_DANA
25.	REKOMENDASISAW

Kelas ini merupakan bentuk model dari tabel STATUS_MAHASISWA yang dipresentasikan dalam kelas STATUS_MAHASISWA.cs. Berikut akan ditunjukkan kode dari kelas model STATUS_MAHASISWA pada Kode Sumber 4.1.

```
namespace TA_Juara.Models
{
    using System;
    using System.Collections.Generic;

    public partial class STATUS_MAHASISWA
    {
        public STATUS_MAHASISWA()
        {
            this.MAHASISWAs = new HashSet<MAHASISWA>();
        }

        public int ID_STATUS_MHS { get; set; }
        public string NAMA_STATUS_MHS { get; set; }

        public virtual ICollection<MAHASISWA> MAHASISWAs
        { get; set; }
    }
}
```

Kode Sumber 4.1 Kelas Model STATUS_MAHASISWA

Untuk semua kelas model hasil generate dari ADO.NET Entity Framework 5 akan mempunyai struktur kelas yang sama menyesuaikan dengan rancangan *database*.

4.2.2. Implementasi View

Lapisan antarmuka merupakan lapisan aplikasi yang berhubungan dengan interaksi aplikasi dengan pengguna. Pada lapisan ini terdapat kelas yang berhubungan dengan tampilan dan aksi yang bersesuaian.

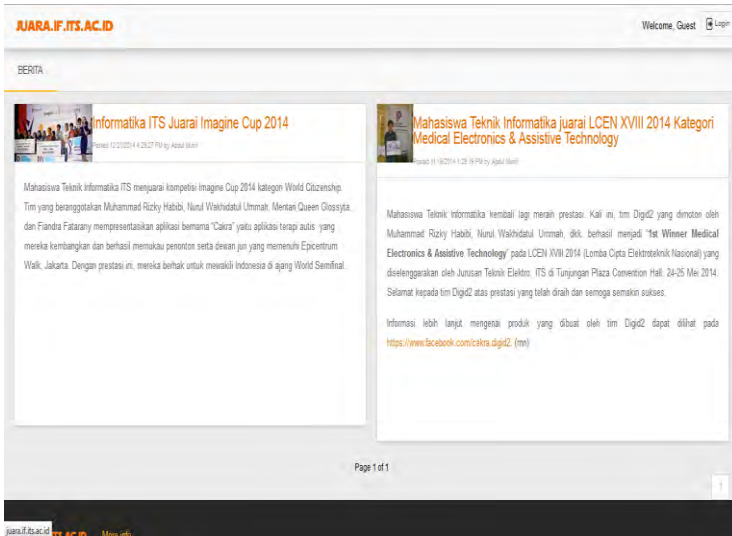
4.2.2.1. Implementasi Master Page

Master Page adalah halaman yang digunakan sebagai template untuk antarmuka halaman lainnya. *Master Page* pada

sistem ini terbagi menjadi 4 bagian yaitu, DashboardKosong, DashboardDosen, Home, dan DashboardMahasiswaKosong. *Master Page* DashboardKosong digunakan sebagai template untuk memanggil kelas **Cascading Style Sheets (CSS)**, kelas **JavaScript**, serta untuk mengatur **SESSION Login**. *Master Page* DashboardDosen digunakan untuk mengatur menu yang dapat diakses oleh pengguna Dosen. *Master Page* Home digunakan untuk mengatur menu untuk pengguna umum. *Master Page* DashboardMahasiswaKosong digunakan untuk mengatur menu pada dashboard mahasiswa.

4.2.2.2. Implementasi Antarmuka Home

Halaman Home seperti pada Gambar 4.1 dan Gambar 4.2 merupakan halaman utama pada aplikasi JUARA yang berbasis website. Halaman ini akan muncul pertama kali saat pengguna dan admin membuka halaman JUARA. Untuk masuk ke halaman admin, administrator perlu melakukan login. Sehingga hanya pengguna yang memiliki akun administrator yang dapat masuk pada halaman admin website ini. Dalam website JUARA ini terbagi menjadi 3 administrator yaitu admin master, admin permohonan dana, dan admin berita. Admin master mempunyai akses untuk mengelola keseluruhan data yang terdapat pada website JUARA. Admin permohonan dana hanya mempunyai hak akses untuk mengelola permohonan dana. Sedangkan untuk admin berita hanya mempunyai untuk mengelola berita. Login juga harus dilakukan oleh mahasiswa yang akan memasuki halaman mahasiswa. Login mahasiswa dilakukan dengan menggunakan username NRP dan password default NRP.



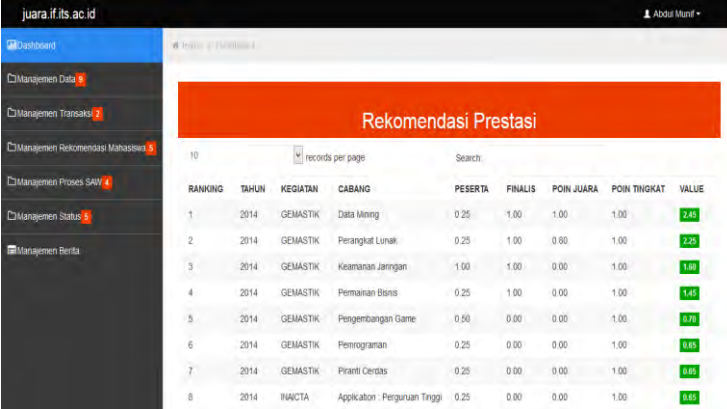
Gambar 4.1 Antarmuka Halaman Utama



Gambar 4.2 Tampilan Berita Pada Halaman Home

4.2.2.3. Implementasi Antarmuka Halaman Dashboard Admin

Antarmuka halaman dashboard admin ini terbagi menjadi tiga jenis yaitu admin master, admin permohonan dana, dan admin berita. Setiap admin mempunyai hak akses yang berbeda. Antarmuka halaman dashboard admin master ditunjukkan pada Gambar 4.3.



The screenshot shows the Admin Dashboard interface. On the left is a sidebar menu with options like 'Manajemen Data', 'Manajemen Transaksi', 'Manajemen Rekomendasi Mahasiswa', 'Manajemen Proses SAV', 'Manajemen Status', and 'Manajemen Berita'. The main content area is titled 'Rekomendasi Prestasi' and displays a table with 8 rows of data. The table includes columns for Ranking, Tahun, Kegiatan, Cabang, Peserta, Finalis, Poin Juara, Poin Tingkat, and Value. Each row represents a recommendation entry with specific details and a value score.

RANKING	TAHUN	KEGIATAN	CABANG	PESERTA	FINALIS	POIN JUARA	POIN TINGKAT	VALUE
1	2014	GEMASTIK	Data Mining	0.25	1.00	1.00	1.00	2.45
2	2014	GEMASTIK	Perangkat Lunak	0.25	1.00	0.00	1.00	2.25
3	2014	GEMASTIK	Keselamatan Jaringan	1.00	1.00	0.00	1.00	1.00
4	2014	GEMASTIK	Pernikahan Bisnis	0.25	1.00	0.00	1.00	1.45
5	2014	GEMASTIK	Pengembangan Game	0.50	0.00	0.00	1.00	0.70
6	2014	GEMASTIK	Pemrograman	0.25	0.00	0.00	1.00	0.05
7	2014	GEMASTIK	Piranti Cerdas	0.25	0.00	0.00	1.00	0.05
8	2014	IWACTA	Application Pengukuran Tinggi	0.25	0.00	0.00	1.00	0.05

Gambar 4.3 Antarmuka Halaman Utama Dashboard Dosen

Halaman utama dashboard admin master menampilkan daftar rekomendasi prestasi yang dilengkapi tombol update, tanggal terakhir update data rekomendasi prestasi, dan label tanda data baru yang masuk dalam proses perhitungan rekomendasi prestasi. Tombol update digunakan untuk menampilkan rekomendasi berdasarkan data terbaru. Jika terdapat data baru masuk dalam proses perhitungan rekomendasi prestasi, hasil rekomendasi prestasi yang baru tidak langsung ditampilkan dalam website. Namun data tersebut disimpan dalam sebuah tabel, hal ini dilakukan untuk mengurangi beban kerja aplikasi. Tanggal terakhir update data rekomendasi prestasi digunakan untuk membantu admin master mengetahui waktu terakhir melakukan update data rekomendasi prestasi. Sedangkan label

tanda data baru digunakan untuk membantu admin mengetahui jika terdapat data baru dalam proses perhitungan rekomendasi prestasi.

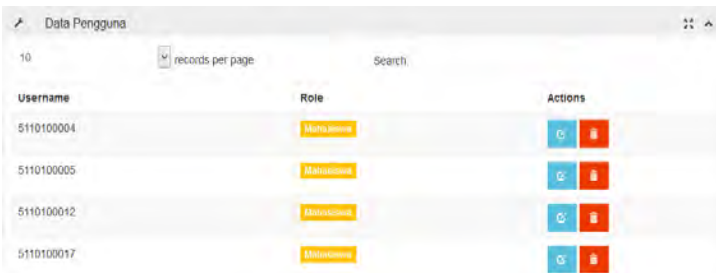
Halaman utama dashboard admin master juga menampilkan 5 daftar prestasi terbaru yang mengalami update. Pada halaman dashboard admin master ini terdapat 6 jenis manajemen, dimana setiap manajemen mempunyai beberapa menu. Berikut akan dijabarkan menu-menu untuk setiap sub manajemen menu.

1. Manajemen Data

Manajemen data ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data yang menjadi kebutuhan aplikasi.

a. Antarmuka Halaman Data Pengguna

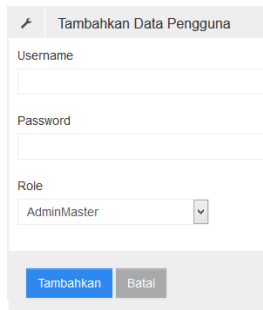
Antarmuka halaman data pengguna ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data pengguna website. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar pengguna dan bagian *form* menambah dan mengubah data anggota. Data pengguna ini terdiri dari *username* pengguna dan *role*. *Username* pengguna menggunakan NRP untuk pengguna dengan *role* mahasiswa, dan NIP untuk *role* admin, hal ini dikarenakan yang berperan sebagai admin adalah dosen. Antarmuka halaman data pengguna dapat dilihat pada Gambar 4.4.



Username	Role	Actions
5110100004	Mahasiswa	[Edit] [Delete] [Add]
5110100005	Mahasiswa	[Edit] [Delete] [Add]
5110100012	Dosen	[Edit] [Delete] [Add]
5110100017	Mahasiswa	[Edit] [Delete] [Add]

Gambar 4.4 Antarmuka Halaman Daftar Pengguna

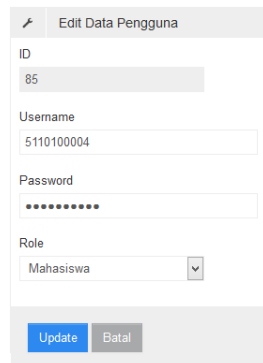
Jika ingin menambahkan data pengguna, maka admin master harus memasukkan *username*, *password*, dan memilih *role* untuk pengguna. Terdapat 4 pilihan *role* yaitu mahasiswa, admin master, admin berita, dan admin permohonan dana. Form untuk menambahkan data pengguna ditunjukkan pada Gambar 4.5.



The form is titled 'Tambahkan Data Pengguna' with a pencil icon. It contains three input fields: 'Username' (text), 'Password' (text), and 'Role' (dropdown menu). The 'Role' dropdown is currently set to 'AdminMaster'. At the bottom, there are two buttons: 'Tambahkan' (blue) and 'Batal' (grey).

Gambar 4.5 Form untuk Menambah Data Pengguna

Pada halaman data pengguna, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data pengguna akan menampilkan *form* untuk *edit* data pengguna. Form edit ini ditunjukkan pada Gambar 4.6.




The form is titled 'Edit Data Pengguna' with a pencil icon. It contains four input fields: 'ID' (text, value: 85), 'Username' (text, value: 5110100004), 'Password' (text, masked with dots), and 'Role' (dropdown menu, value: Mahasiswa). At the bottom, there are two buttons: 'Update' (blue) and 'Batal' (grey).

Gambar 4.6 Form untuk Edit Data Pengguna

b. Antarmuka Halaman Data Mahasiswa

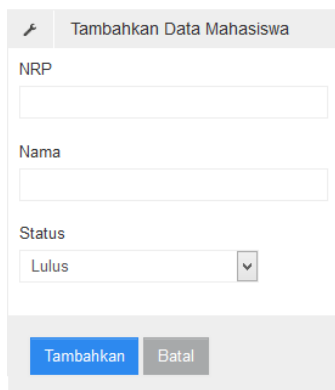
Antarmuka halaman data mahasiswa ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data mahasiswa. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar mahasiswa dan bagian *form* menambah dan mengubah data mahasiswa. Data mahasiswa ini terdiri dari NRP dan nama mahasiswa. Antarmuka halaman data mahasiswa diperlihatkan pada Gambar 4.7.



NRP	Nama	Status	Actions
5110100001	GEMBONG SATRIA GELAR BAYA	Mahasiswa	 
5110100004	IKHWAN FARISANDANI	Mahasiswa	 
5110100005	RAMADHAN SATYA PUTRA	Mahasiswa	 

Gambar 4.7 Isi Halaman Data Mahasiswa

Jika ingin menambahkan data mahasiswa, maka admin master harus memasukan NRP, nama, dan memilih status. Terdapat 2 pilihan status yaitu mahasiswa dan lulus. Form untuk menambahkan data mahasiswa ditunjukkan pada Gambar 4.8.



Tambahkan Data Mahasiswa

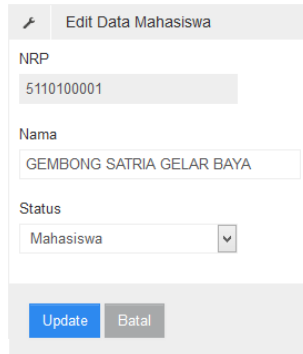
NRP

Nama

Status

Gambar 4.8 Form untuk Menambah Data Mahasiswa

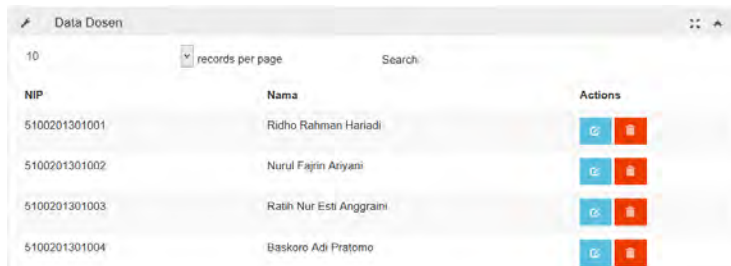
Pada halaman data mahasiswa, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data mahasiswa akan menampilkan *form* untuk *edit* data mahasiswa. Form edit ini ditunjukkan pada Gambar 4.9.











Gambar 4.9 Form untuk Edit Data Mahasiswa

c. Antarmuka Halaman Data Dosen

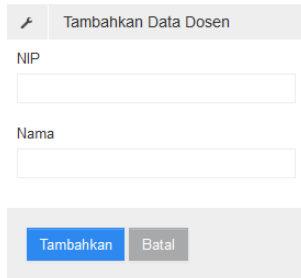
Antarmuka halaman data dosen ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data dosen. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar dosen dan bagian *form* menambah dan mengubah data dosen. Data dosen ini terdiri dari NIP dan nama dosen. Antarmuka halaman data dosen diperlihatkan pada Gambar 4.10.



NIP	Nama	Actions
5100201301001	Ridho Rahman Hariadi	 
5100201301002	Nurul Fajrin Ariyanti	 
5100201301003	Ratih Nur Esti Anggraini	 
5100201301004	Baskoro Adi Pratomo	 

Gambar 4.10 Antarmuka Halaman Data Dosen

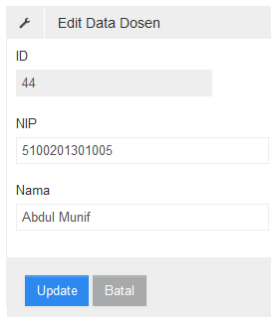
Jika ingin menambahkan data dosen, maka admin master harus memasukkan NIP dan nama. Form untuk menambahkan data mahasiswa ditunjukkan pada Gambar 4.11.



The form is titled 'Tambahkan Data Dosen' with a key icon. It contains two text input fields: 'NIP' and 'Nama'. At the bottom, there are two buttons: 'Tambahkan' (blue) and 'Batal' (grey).

Gambar 4.11 Form untuk Menambah Data Dosen

Pada halaman data dosen, admin dapat melakukan perubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data dosen akan menampilkan *form* untuk *edit* data dosen. Form edit ini ditunjukkan pada Gambar 4.12.



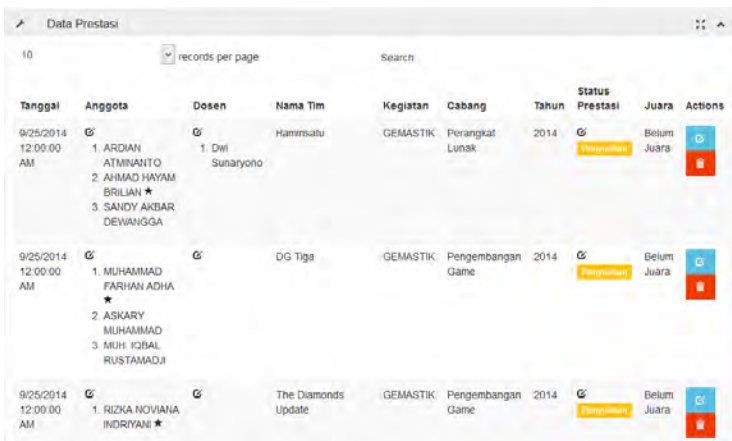
The form is titled 'Edit Data Dosen' with a key icon. It contains four text input fields: 'ID' (pre-filled with '44'), 'NIP' (pre-filled with '5100201301005'), and 'Nama' (pre-filled with 'Abdul Munif'). At the bottom, there are two buttons: 'Update' (blue) and 'Batal' (grey).

Gambar 4.12 Form untuk Edit Data Pengguna







d. Antarmuka Halaman Data Prestasi

Antarmuka halaman data prestasi ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data prestasi. Data prestasi terdiri dari tanggal terakhir prestasi

diperbarui, nama anggota tim, nama dosen pembimbing, nama tim, nama kegiatan, nama cabang kegiatan, tahun kegiatan, status prestasi, dan tingkat juara. Pada kolom anggota, dosen pembimbing, status prestasi, dan juara terdapat *icon* untuk melakukan *edit*. Dimana anggota tim admin master dapat menambah nama anggota sejumlah jumlah anggota yang dimasukan. Nama dosen pembimbing dapat ditambahkan sesuai dengan kebutuhan. Status prestasi dapat diubah jika data prestasi mengalami perubahan tahap, misalkan dari tahap peserta menjadi tahap penyisihan. Untuk juara sedikit berbeda dengan lainnya, *icon edit* akan muncul pada kolom juara jika status prestasi telah berubah menjadi juara. Antarmuka halaman data prestasi diperlihatkan pada Gambar 4.13.

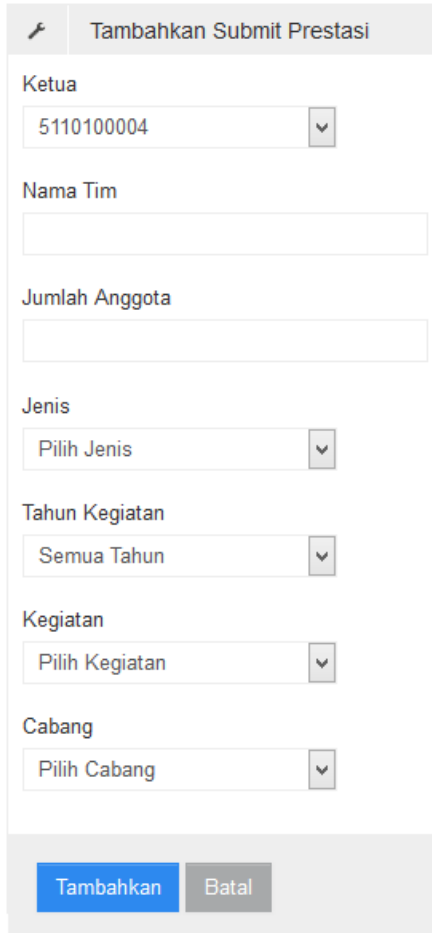



The screenshot shows a web application titled "Data Prestasi". It features a table with columns: Tanggal, Anggota, Dosen, Nama Tim, Kegiatan, Cabang, Tahun, Status Prestasi, Juara, and Actions. The table contains three rows of data, each representing a student's achievement. The first row shows a student named ARDIAN ATMINANTO with a status of "Peserta" and a trophy icon. The second row shows a student named MUHAMMAD FARHAN ADHA with a status of "Peserta" and a trophy icon. The third row shows a student named RIZKA NOVIANA INDIRYANI with a status of "Peserta" and a trophy icon. The interface also includes a search bar, a dropdown for "records per page" set to 10, and a "Search" button.

Tanggal	Anggota	Dosen	Nama Tim	Kegiatan	Cabang	Tahun	Status Prestasi	Juara	Actions
9/25/2014 12:00:00 AM	1. ARDIAN ATMINANTO 2. AHMAD HAYAM BRILIAN ★ 3. SANDY AKBAR DEWANGGA	1. Dwi Sunaryono	Hamirsatu	GEMASTIK	Perangkat Lunak	2014	Peserta	Belum Juara	 
9/25/2014 12:00:00 AM	1. MUHAMMAD FARHAN ADHA ★ 2. ASKARY MUHAMMAD 3. MUH. ICBAL RUSTAMADJI		DG Tiga	GEMASTIK	Pengembangan Game	2014	Peserta	Belum Juara	 
9/25/2014 12:00:00 AM	1. RIZKA NOVIANA INDIRYANI ★		The Diamonds Update	GEMASTIK	Pengembangan Game	2014	Peserta	Belum Juara	 

Gambar 4.13 Antarmuka Halaman Data Prestasi

Jika ingin menambahkan data prestasi, maka admin master harus memilih NRP ketua, memasukan nama tim, jumlah anggota, memilih jenis kegiatan, tahun kegiatan, kegiatan, dan cabang kegiatan. Form untuk menambahkan data mahasiswa ditunjukan pada Gambar 4.14.



 Tambahkan Submit Prestasi

Ketua
5110100004

Nama Tim

Jumlah Anggota

Jenis
Pilih Jenis

Tahun Kegiatan
Semua Tahun


Kegiatan
Pilih Kegiatan

Cabang
Pilih Cabang

Tambahkan Batal

Gambar 4.14 Form untuk Menambah Data Prestasi

Pada halaman data prestasi, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data prestasi akan menampilkan *form* untuk *edit* data prestasi. Form edit ini ditunjukkan pada Gambar 4.15.

 Edit Submit Prestasi

ID

2

ID Pengguna

174

Nama Tim

Haminsatu

Jumlah Anggota

3

Jenis

Lomba

▼

Tahun Kegiatan

2014

▼

Kegiatan

GEMASTIK

▼

Cabang

Perangkat Lunak

▼

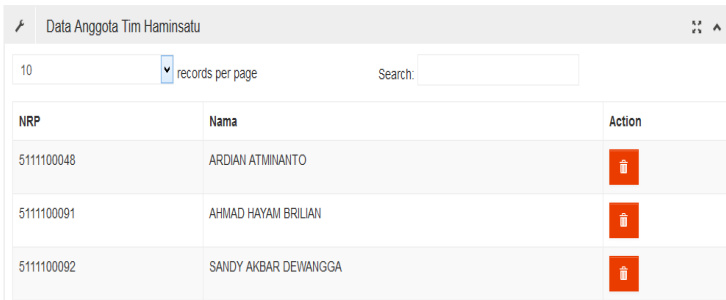
Update




Batal

Gambar 4.15 Form untuk Edit Data Prestasi

Data anggota pada data prestasi dapat diperbarui dengan melakukan klik pada *icon edit* yang terletak dalam kolom anggota. Kemudian akan muncul antarmuka halaman tambah anggota yang terbagi menjadi dua bagian yaitu tabel daftar

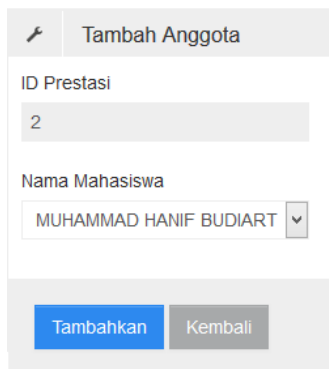
anggota dan *form* untuk menambahkan anggota. Gambar Gambar 4.16 merupakan halaman tambah anggota yang menampilkan data anggota yang terdiri dari NRP dan nama.



NRP	Nama	Action
5111100048	ARDIAN ATMINANTO	
5111100091	AHMAD HAYAM BRILIAN	
5111100092	SANDY AKBAR DEWANGGA	

Gambar 4.16 Antarmuka Halaman Tambah Anggota

Jika admin master ingin menambahkan anggota maka admin master hanya perlu memilih nama mahasiswa yang telah tersimpan dalam sistem. Form untuk tambah anggota dapat dilihat pada Gambar 4.17.



Tambah Anggota

ID Prestasi

2

Nama Mahasiswa

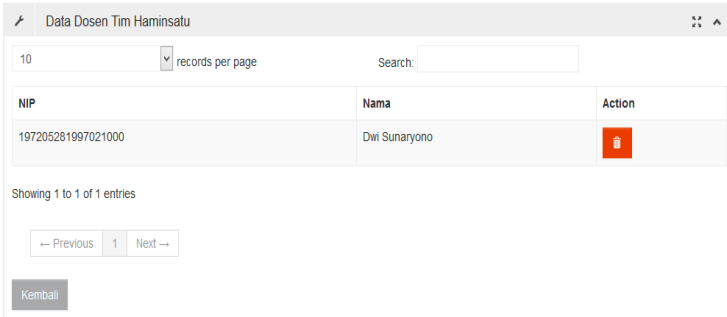
MUHAMMAD HANIF BUDIART

Tambahkan Kembali


Gambar 4.17 Form untuk Tambah Anggota

Data dosen pada info prestasi dapat diperbarui dengan melakukan klik pada *icon edit* yang terletak dalam kolom dosen. Kemudian akan muncul antarmuka halaman tambah dosen

yang terbagi menjadi dua bagian yaitu tabel daftar dosen dan *form* untuk menambahkan dosen. Gambar 4.18 merupakan halaman tambah dosen yang menampilkan data anggota yang terdiri dari NIP dan nama.



The screenshot shows a web interface titled "Data Dosen Tim Haminsatu". It features a search bar and a table with columns "NIP", "Nama", and "Action". The table contains one entry for NIP 197205281997021000 and Nama Dwi Sunaryono. Below the table, it says "Showing 1 to 1 of 1 entries" and includes pagination controls. A "Kembali" button is at the bottom left.

NIP	Nama	Action
197205281997021000	Dwi Sunaryono	

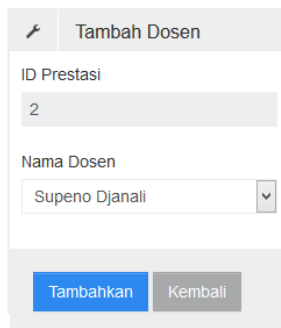
Showing 1 to 1 of 1 entries

← Previous 1 Next →

Kembali

Gambar 4.18 Antarmuka Halaman Tambah Dosen

Jika admin master ingin menambahkan dosen maka admin master hanya perlu memilih nama dosen yang telah tersimpan dalam sistem. Form untuk tambah dosen dapat dilihat pada Gambar 4.19.



The screenshot shows a form titled "Tambah Dosen". It has two input fields: "ID Prestasi" with the value "2" and "Nama Dosen" with a dropdown menu showing "Supeno Djanali". At the bottom, there are two buttons: "Tambahkan" (blue) and "Kembali" (grey).

Tambah Dosen

ID Prestasi

2

Nama Dosen

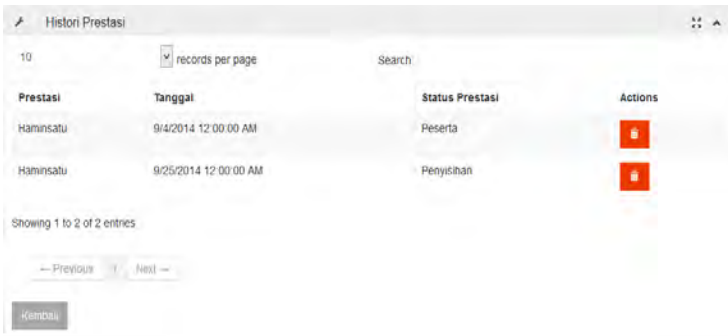
Supeno Djanali

Tambahkan Kembali

Gambar 4.19 Form untuk Tambah Dosen

Status prestasi pada info prestasi dapat diperbarui dengan melakukan klik pada *icon edit* yang terletak dalam kolom status prestasi. Kemudian akan muncul antarmuka

halaman histori prestasi untuk *update* status prestasi yang terbagi menjadi dua bagian yaitu tabel daftar histori status prestasi dan *form* untuk memperbarui status prestasi. Gambar 4.20 merupakan halaman *update* histori prestasi yang menampilkan daftar prestasi yang terdiri dari nama tim, tanggal *update* status prestasi, dan status prestasi.



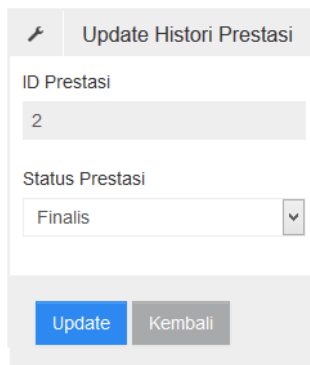
The screenshot shows a web interface titled 'Histori Prestasi'. It includes a search bar, a 'records per page' dropdown set to '10', and a table with the following data:

Prestasi	Tanggal	Status Prestasi	Actions
Haminsatu	9/4/2014 12:00:00 AM	Peserta	
Haminsatu	9/25/2014 12:00:00 AM	Penyisihan	

Below the table, it says 'Showing 1 to 2 of 2 entries'. At the bottom, there are navigation links: 'Previous', '1', 'Next', and a 'Kembali' button.

Gambar 4.20 Antarmuka Halaman Histori Prestasi

Jika admin master ingin memperbarui status prestasi maka admin master hanya perlu memilih status prestasi yang telah tersimpan dalam sistem. Form untuk *update* status prestasi dapat dilihat pada Gambar 4.21.



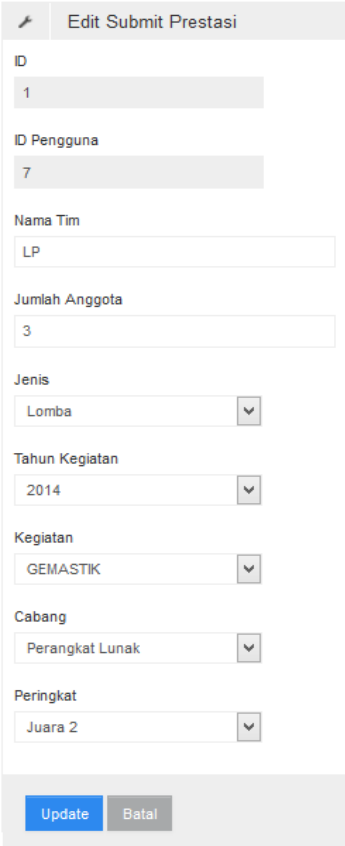
The screenshot shows a form titled 'Update Histori Prestasi'. It contains the following fields:

- ID Prestasi:** A text input field containing the value '2'.
- Status Prestasi:** A dropdown menu with 'Finalis' selected.

At the bottom of the form, there are two buttons: 'Update' (in blue) and 'Kembali' (in grey).

Gambar 4.21 Form untuk *Update* Status Prestasi

Juara pada info prestasi dapat diperbarui jika status prestasi tim telah menjadi juara. Juara ini dapat diperbarui dengan melakukan klik pada *icon edit* yang terletak dalam kolom *Actions*. Kemudian akan muncul *form edit* info prestasi dengan tambahan *dropdown* untuk *update* peringkat. Gambar 4.22 menunjukkan *form edit* info prestasi dengan tambahan *dropdown update* peringkat.



Edit Submit Prestasi

ID
1

ID Pengguna
7

Nama Tim
LP

Jumlah Anggota
3

Jenis
Lomba

Tahun Kegiatan
2014

Kegiatan
GEMASTIK

Cabang
Perangkat Lunak

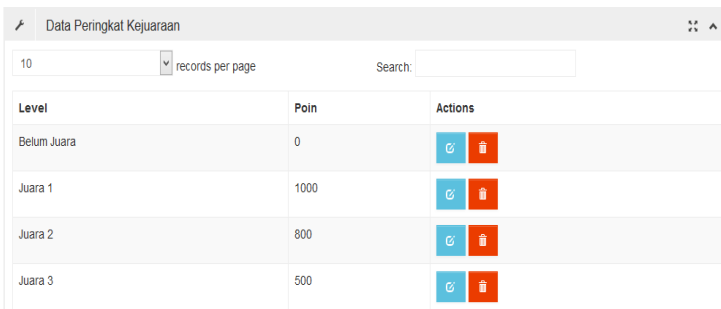
Peringkat
Juara 2

Update **Batal**

Gambar 4.22 Antarmuka Halaman Form Edit Info Prestasi dengan Peringkat

e. Antarmuka Halaman Data Peringkat Kejuaraan

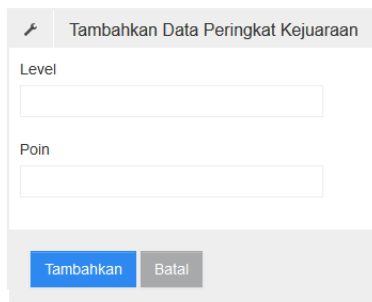
Antarmuka halaman data peringkat kejuaraan ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data peringkat kejuaraan. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar peringkat kejuaraan dan bagian *form* menambah dan mengubah data peringkat kejuaraan. Data peringkat kejuaraan ini terdiri dari level dan poin. Antarmuka halaman data peringkat kejuaraan diperlihatkan pada Gambar 4.23.



Level	Poin	Actions
Belum Juara	0	
Juara 1	1000	
Juara 2	800	
Juara 3	500	

Gambar 4.23 Antarmuka Halaman Data Peringkat Kejuaraan

Jika ingin menambahkan data peringkat kejuaraan, maka admin master harus memasukan level dan poin. Form untuk menambahkan data mahasiswa ditunjukkan pada Gambar 4.24.



Tambahkan Data Peringkat Kejuaraan

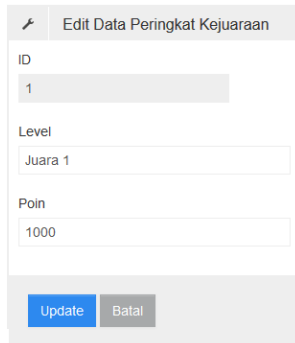
Level

Poin

Tambahkan Batal

Gambar 4.24 Form untuk Menambah Data Peringkat Kejuaraan

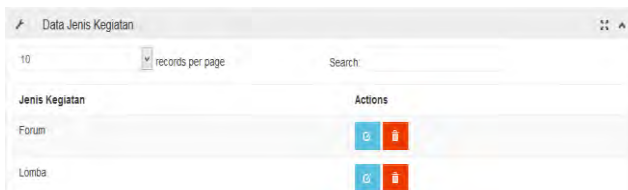
Pada halaman data peringkat kejuaraan, admin dapat melakukan perubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data peringkat kejuaraan akan menampilkan *form* untuk *edit* data peringkat kejuaraan. Form edit ini ditunjukkan pada Gambar 4.25.







Gambar 4.25 Form untuk Edit Data Pengguna

f. Antarmuka Halaman Data Jenis Kegiatan

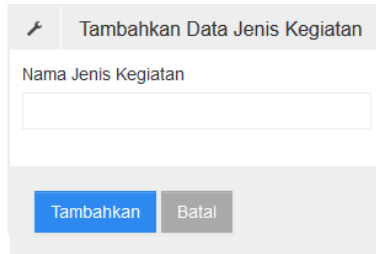
Antarmuka halaman data jenis kegiatan ini digunakan untuk melakukan penambahan, perubahan, dan penghapusan data jenis kegiatan. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar tingkat kejuaraan dan bagian *form* menambah dan mengubah data tingkat kejuaraan. Data tingkat kejuaraan ini terdiri dari level dan poin. Antarmuka halaman data tingkat kejuaraan diperlihatkan pada Gambar 4.26.



Jenis Kegiatan	Actions
Forum	 
Lomba	 

Gambar 4.26 Antarmuka Halaman Data Jenis Kegiatan

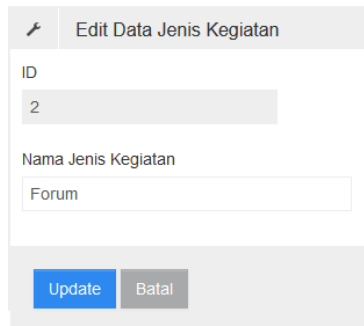
Jika ingin menambahkan data jenis kegiatan, maka admin master hanya perlu memasukan nama jenis kegiatan. Form untuk menambahkan data jenis kegiatan ditunjukkan pada Gambar 4.27.



The form is titled 'Tambahkan Data Jenis Kegiatan' with a wrench icon. It contains a text input field labeled 'Nama Jenis Kegiatan'. At the bottom, there are two buttons: 'Tambahkan' (blue) and 'Batal' (grey).

Gambar 4.27 Form untuk Menambah Data Jenis Kegiatan

Pada halaman data jenis kegiatan, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data jenis kegiatan akan menampilkan *form* untuk *edit* data jenis kegiatan. Form edit ini ditunjukkan pada Gambar 4.28.



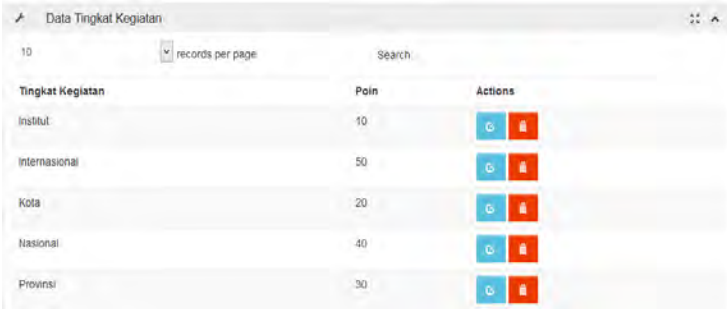
The form is titled 'Edit Data Jenis Kegiatan' with a wrench icon. It contains two text input fields: 'ID' with the value '2' and 'Nama Jenis Kegiatan' with the value 'Forum'. At the bottom, there are two buttons: 'Update' (blue) and 'Batal' (grey).

Gambar 4.28 Form untuk Edit Data Jenis Kegiatan

g. Antarmuka Halaman Data Tingkat Kegiatan

Antarmuka halaman data tingkat kegiatan ini digunakan untuk melakukan penambahan, pengubahan, dan

penghapusan data tingkat kegiatan. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar tingkat kegiatan dan bagian *form* menambah dan mengubah data tingkat kegiatan. Data tingkat kegiatan ini terdiri dari tingkat kegiatan dan poin. Antarmuka halaman data tingkat kegiatan diperlihatkan pada Gambar 4.29.

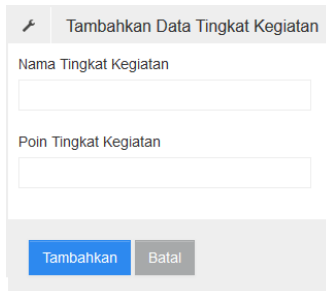


The screenshot shows a web interface titled "Data Tingkat Kegiatan". It features a table with three columns: "Tingkat Kegiatan", "Poin", and "Actions". The table contains five rows of data. Above the table, there is a "records per page" dropdown set to "10" and a "Search" input field.

Tingkat Kegiatan	Poin	Actions
Instansi	10	[Edit] [Delete]
Internasional	50	[Edit] [Delete]
Kota	20	[Edit] [Delete]
Nasional	40	[Edit] [Delete]
Provinsi	30	[Edit] [Delete]

Gambar 4.29 Antarmuka Halaman Data Tingkat Kegiatan

Jika ingin menambahkan data tingkat kegiatan, maka admin master harus memasukan nama tingkat kegiatan dan poin tingkat kegiatan. Form untuk menambahkan data tingkat kegiatan ditunjukkan pada Gambar 4.30.

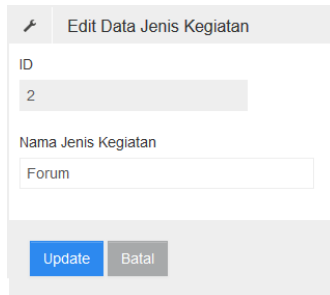


The screenshot shows a form titled "Tambahkan Data Tingkat Kegiatan". It has two input fields: "Nama Tingkat Kegiatan" and "Poin Tingkat Kegiatan". At the bottom, there are two buttons: "Tambahkan" (Add) and "Batal" (Cancel).

Gambar 4.30 Form untuk Menambah Data Tingkat Kegiatan

Pada halaman data tingkat kegiatan, admin dapat melakukan perubahan data dengan memilih *icon edit*

berwarna biru pada kolom *actions*. Kemudian halaman data tingkat kegiatan akan menampilkan *form* untuk *edit* data tingkat kegiatan. Form edit ini ditunjukkan pada Gambar 4.31.

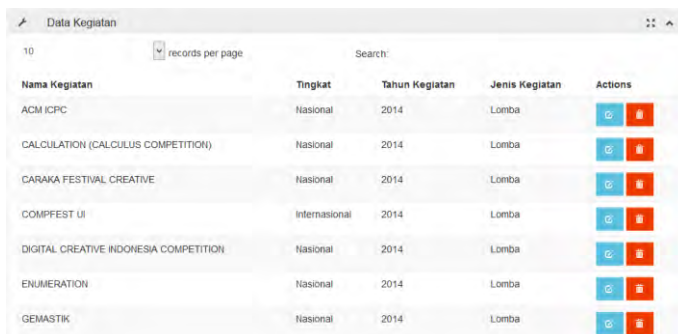


The image shows a form titled "Edit Data Jenis Kegiatan". It contains a text input field for "ID" with the value "2", and another text input field for "Nama Jenis Kegiatan" with the value "Forum". At the bottom, there are two buttons: "Update" (blue) and "Batal" (grey).















Gambar 4.31 Form untuk Edit Data Tingkat Kegiatan

h. Antarmuka Halaman Data Kegiatan

Antarmuka halaman data kegiatan ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data kegiatan. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar kegiatan dan bagian *form* menambah dan mengubah data kegiatan. Data kegiatan ini terdiri dari nama kegiatan, tingkat, tahun kegiatan, dan jenis kegiatan. Antarmuka halaman data kegiatan diperlihatkan pada Gambar 4.32.

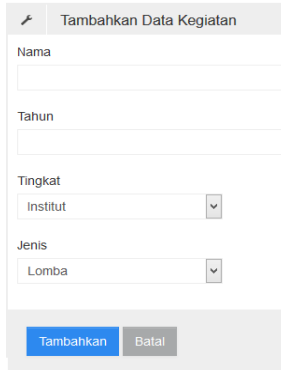


The image shows a web interface for "Data Kegiatan". It includes a search bar, a "records per page" dropdown set to "10", and a table with the following data:

Nama Kegiatan	Tingkat	Tahun Kegiatan	Jenis Kegiatan	Actions
ACM ICPC	Nasional	2014	Lomba	 
CALCULATION (CALCULUS COMPETITION)	Nasional	2014	Lomba	 
CARAKA FESTIVAL CREATIVE	Nasional	2014	Lomba	 
COMPFEST UI	Internasional	2014	Lomba	 
DIGITAL CREATIVE INDONESIA COMPETITION	Nasional	2014	Lomba	 
ENUMERATION	Nasional	2014	Lomba	 
GEMASTIK	Nasional	2014	Lomba	 

Gambar 4.32 Antarmuka Halaman Data Kegiatan

Jika ingin menambahkan data kegiatan, maka admin master harus memasukan nama dan tahun, serta memilih tingkat kegiatan dan jenis kegiatan. Form untuk menambahkan data kegiatan ditunjukan pada Gambar 4.33.



The screenshot shows a web form titled 'Tambahkan Data Kegiatan' with a pencil icon. It contains four input fields: 'Nama' (text), 'Tahun' (text), 'Tingkat' (dropdown menu with 'Institut' selected), and 'Jenis' (dropdown menu with 'Lomba' selected). At the bottom are two buttons: 'Tambahkan' (blue) and 'Batal' (grey).

Gambar 4.33 Form untuk Menambah Data Kegiatan

Pada halaman data kegiatan, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data kegiatan akan menampilkan *form* untuk *edit* data kegiatan. Form edit ini ditunjukan pada Gambar 4.34.

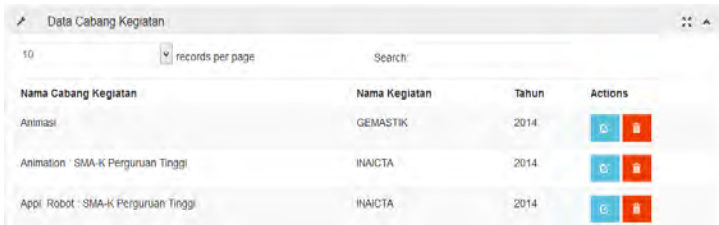








The screenshot shows a web form titled 'Edit Data Kegiatan' with a pencil icon. It contains five input fields: 'ID' (text with value '5'), 'Nama' (text with value 'ACM ICPC'), 'Tahun' (text with value '2014'), 'Tingkat' (dropdown menu with 'Nasional' selected), and 'Jenis' (dropdown menu with 'Lomba' selected). At the bottom are two buttons: 'Update' (blue) and 'Batal' (grey).

Gambar 4.34 Form untuk Edit Data Kegiatan

i. Antarmuka Halaman Data Cabang Kegiatan

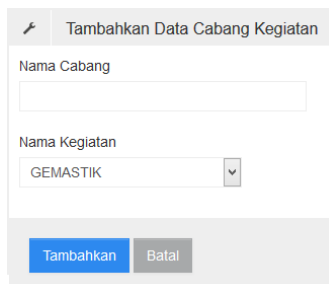
Antarmuka halaman data cabang kegiatan ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data tingkat kegiatan. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar cabang kegiatan dan bagian *form* menambah dan mengubah data cabang kegiatan. Data cabang kegiatan ini terdiri dari nama cabang kegiatan, nama kegiatan, dan tahun kegiatan. Antarmuka halaman data cabang kegiatan diperlihatkan pada Gambar 4.35.



Nama Cabang Kegiatan	Nama Kegiatan	Tahun	Actions
Animasi	GEMASTIK	2014	 
Animation : SMA-K Perguruan Tinggi	INAICTA	2014	 
Appl. Robot : SMA-K Perguruan Tinggi	INAICTA	2014	 

Gambar 4.35 Antarmuka Halaman Data Cabang Kegiatan

Jika ingin menambahkan data cabang kegiatan, maka admin master harus memasukkan nama cabang dan memilih nama kegiatan. Form untuk menambahkan data cabang kegiatan ditunjukkan pada Gambar 4.36.



Tambahkan Data Cabang Kegiatan

Nama Cabang

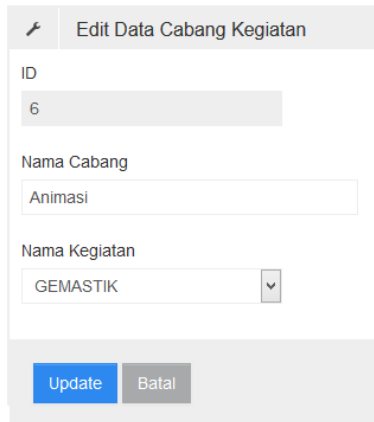
Nama Kegiatan

GEMASTIK

Tambahkan Batal

Gambar 4.36 Form untuk Menambah Data Cabang Kegiatan

Pada halaman data cabang kegiatan, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data cabang kegiatan akan menampilkan *form* untuk *edit* data cabang kegiatan. Form edit ini ditunjukkan pada Gambar 4.37.



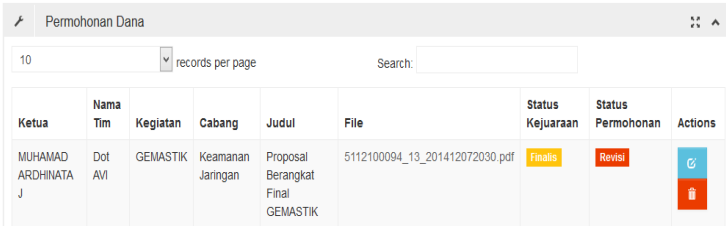
The image shows a web form titled "Edit Data Cabang Kegiatan". The form is contained within a light grey box. At the top of the box is a header bar with a wrench icon on the left and the text "Edit Data Cabang Kegiatan" on the right. Below the header, the form contains three input fields. The first is labeled "ID" and has a text input field with the value "6". The second is labeled "Nama Cabang" and has a text input field with the value "Animasi". The third is labeled "Nama Kegiatan" and has a dropdown menu with the value "GEMASTIK" and a downward arrow. At the bottom of the form, there are two buttons: a blue button labeled "Update" and a grey button labeled "Batal".



Gambar 4.37 Form untuk Edit Data Cabang Kegiatan

2. Manajemen Transaksi

a. Antarmuka Halaman Permohonan Dana

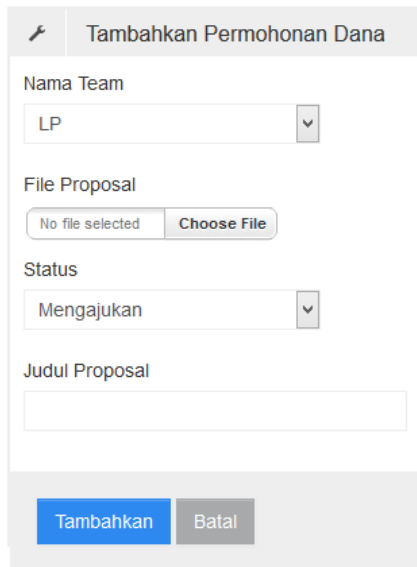
Antarmuka halaman permohonan dana ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan permohonan dana. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar permohonan dana dan bagian *form* menambah dan mengubah permohonan dana. Permohonan dana terdiri dari nama ketua, nama tim, nama kegiatan, cabang kegiatan, judul proposal, *file* proposal, status kejuaraan, dan status permohonan dana. Antarmuka halaman permohonan dana diperlihatkan pada Gambar 4.38.



Ketua	Nama Tim	Kegiatan	Cabang	Judul	File	Status Kejuaraan	Status Permohonan	Actions
MUHAMAD ARDHIHATA J	Dot AVI	GEMASTIK	Keamanan Jaringan	Proposal Berangkat Final GEMASTIK	5112100094_13_201412072030.pdf	Finalis	Revisi	 

Gambar 4.38 Antarmuka Halaman Permohonan Dana

Jika ingin menambahkan permohonan dana, maka admin master harus memilih nama tim, mengunggah *file* proposal, memilih status permohonan dana, dan memasukan judul proposal. Bentuk *file* proposal yang dapat diunggah adalah *file* dalam bentuk *pdf* dan *word*. Sedangkan untuk status permohonan dana terdapat 4 jenis status yaitu OK, revisi, ditolak, dan mengajukan. Form untuk menambahkan permohonan dana ditunjukkan pada Gambar 4.39.



Tambahkan Permohonan Dana

Nama Team

LP

File Proposal

No file selected Choose File

Status

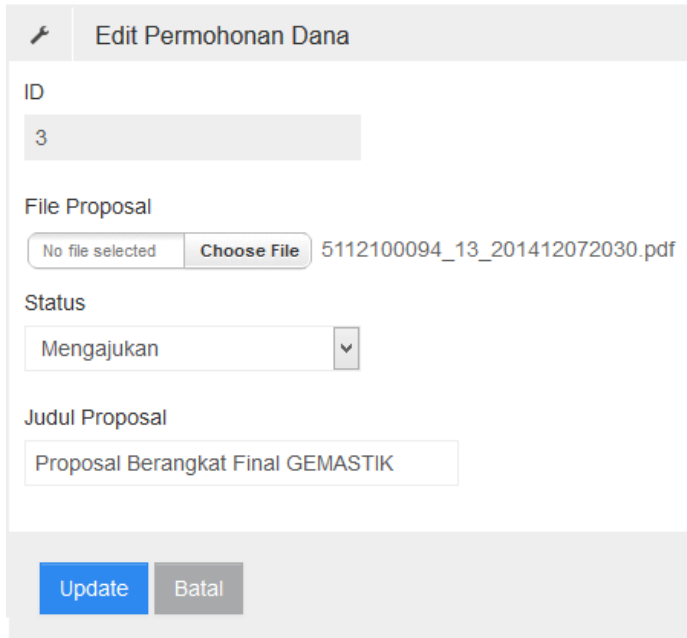
Mengajukan

Judul Proposal

Tambahkan Batal

Gambar 4.39 Form untuk Menambah Permohonan Dana

Pada halaman permohonan dana, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman permohonan dana akan menampilkan *form* untuk *edit* permohonan dana. Form edit ini ditunjukkan pada Gambar 4.40.



Edit Permohonan Dana

ID

3

File Proposal

No file selected Choose File 5112100094_13_201412072030.pdf

Status

Mengajukan

Judul Proposal

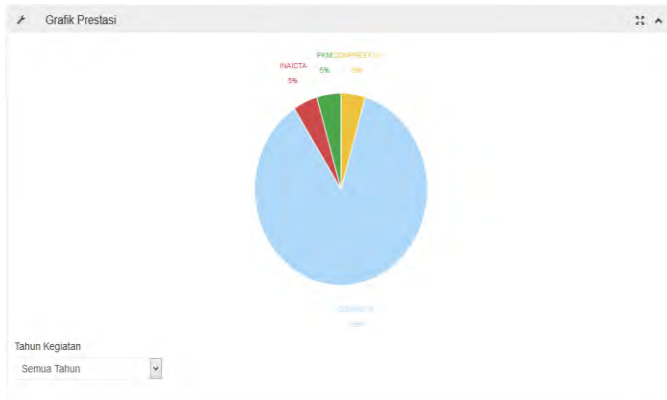
Proposal Berangkat Final GEMASTIK

Update Batal

Gambar 4.40 Form untuk Edit Permohonan Dana

b. Antarmuka Halaman Laporan Prestasi

Antarmuka halaman laporan prestasi menampilkan grafik prestasi dan laporan data prestasi. Antarmuka halaman laporan prestasi dalam bentuk grafik diperlihatkan pada Gambar 4.41.



Gambar 4.41 Antarmuka Grafik Prestasi

Laporan data prestasi ditampilkan dalam bentuk tabel yang memuat nama ketua, nama tim, jumlah anggota, data anggota, dosen pembimbing, nama kegiatan, nama cabang, tahun kegiatan, status prestasi, dan tingkat kejuaraan. Tabel laporan data prestasi ditunjukkan pada Gambar 4.42.

Tanggal	Anggota	Dosen	Nama Tim	Kegiatan	Cabang	Tahun	Status Prestasi	Juara	Actions
9/25/2014 12:00:00 AM	1. ARDIAN ATMINANTO 2. AHMAD HAYAM BRILIAN ★ 3. SANDY AKBAR DEWANOGA	1. Dwi Sunaryono	Haminsatu	GEMASTIK	Perangkat Lunak	2014	Prestasi Baik	Belum Juara	[Edit] [Delete]

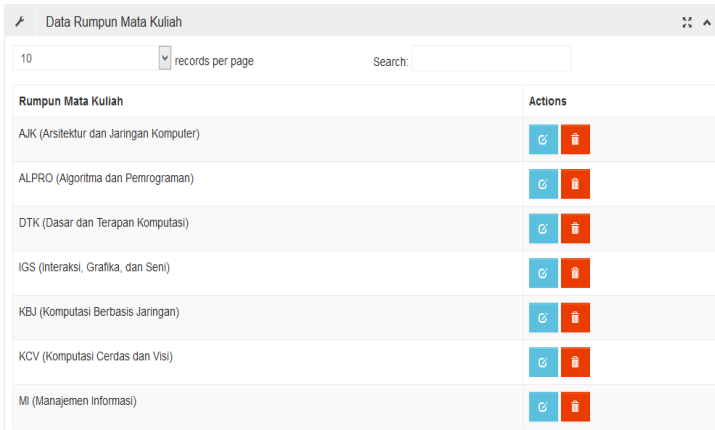
Gambar 4.42 Laporan Prestasi















3. Manajemen Rekomendasi Mahasiswa

a. Antarmuka Halaman Data Rumpun Mata Kuliah

Antarmuka halaman data rumpun mata kuliah ini digunakan untuk melakukan penambahan, perubahan, dan penghapusan data rumpun mata kuliah pada website. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar

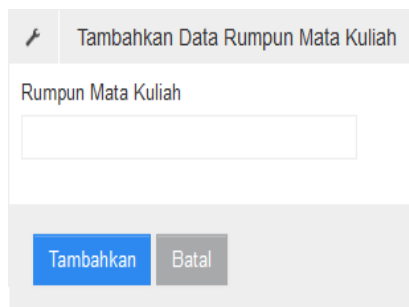
rumpun mata kuliah dan bagian *form* menambah dan mengubah data rumpun mata kuliah. Data pengguna ini terdiri dari nama rumpun mata kuliah. Antarmuka halaman data rumpun mata kuliah dapat dilihat pada Gambar 4.43.




Rumpun Mata Kuliah	Actions
AJK (Arsitektur dan Jaringan Komputer)	 
ALPRO (Algoritma dan Pemrograman)	 
DTK (Dasar dan Terapan Komputasi)	 
IGS (Interaksi, Grafika, dan Seni)	 
KBJ (Komputasi Berbasis Jaringan)	 
KCV (Komputasi Cerdas dan Visi)	 
MI (Manajemen Informasi)	 

Gambar 4.43 Antarmuka Halaman Data Rumpun Mata Kuliah

Jika ingin menambahkan data rumpun mata kuliah, maka admin master harus memasukan nama rumpun mata kuliah. Form untuk menambahkan data rumpun mata kuliah ditunjukkan pada Gambar 4.44.

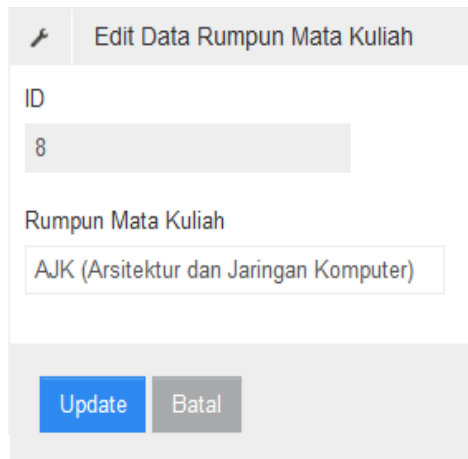


 Tambahkan Data Rumpun Mata Kuliah

Rumpun Mata Kuliah

Gambar 4.44 Form untuk Menambah Data Rumpun Mata Kuliah

Pada halaman data rumpun mata kuliah, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data rumpun mata kuliah akan menampilkan *form* untuk *edit* data rumpun mata kuliah. Form edit ini ditunjukkan pada Gambar 4.45.

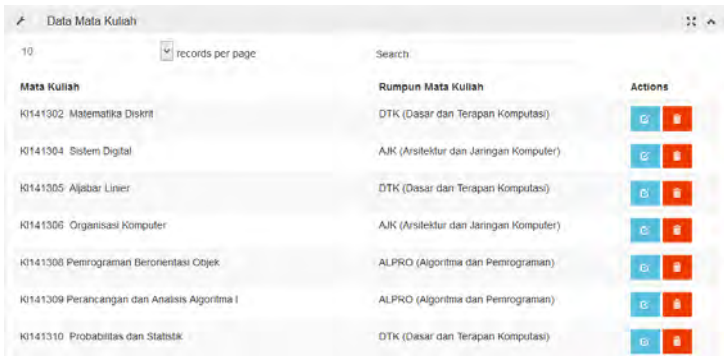
















The image shows a web form titled "Edit Data Rumpun Mata Kuliah". It features a header bar with a wrench icon and the title. Below the header, there are two input fields. The first field is labeled "ID" and contains the number "8". The second field is labeled "Rumpun Mata Kuliah" and contains the text "AJK (Arsitektur dan Jaringan Komputer)". At the bottom of the form, there are two buttons: a blue "Update" button and a grey "Batal" button.

Gambar 4.45 Form untuk Edit Data Rumpun Mata Kuliah

b. Antarmuka Halaman Data Mata Kuliah

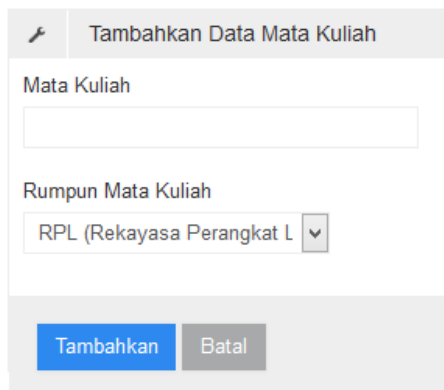
Antarmuka halaman data mata kuliah ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data mata kuliah pada website. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar mata kuliah dan bagian *form* menambah dan mengubah data mata kuliah. Data mata kuliah ini terdiri dari nama mata kuliah. Antarmuka halaman data mata kuliah dapat dilihat pada Gambar 4.46.



Mata Kuliah	Rumpun Mata Kuliah	Actions
KI141302 Matematika Diskrit	DTK (Dasar dan Terapan Komputasi)	 
KI141304 Sistem Digital	AJK (Arsitektur dan Jaringan Komputer)	 
KI141305 Aljabar Linier	DTK (Dasar dan Terapan Komputasi)	 
KI141306 Organisasi Komputer	AJK (Arsitektur dan Jaringan Komputer)	 
KI141308 Pemrograman Berorientasi Objek	ALPRO (Algoritma dan Pemrograman)	 
KI141309 Perancangan dan Analisis Algoritma I	ALPRO (Algoritma dan Pemrograman)	 
KI141310 Probabilitas dan Statistik	DTK (Dasar dan Terapan Komputasi)	 

Gambar 4.46 Antarmuka Halaman Data Mata Kuliah


Jika ingin menambahkan data mata kuliah, maka admin master harus memasukkan nama mata kuliah dan memilih rumpun mata kuliah. Form untuk menambahkan data mata kuliah ditunjukkan pada Gambar 4.47.



Tambahkan Data Mata Kuliah

Mata Kuliah

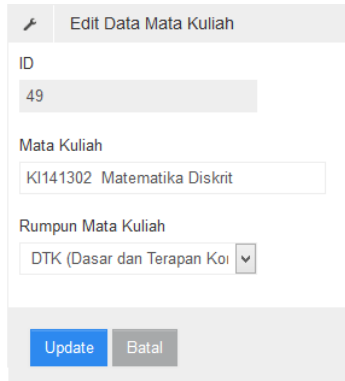
Rumpun Mata Kuliah

RPL (Rekayasa Perangkat L 

Tambahkan **Batal**

Gambar 4.47 Form untuk Menambah Data Mata Kuliah

Pada halaman data mata kuliah, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data mata kuliah akan menampilkan *form* untuk *edit* data mata kuliah. Form *edit* ini ditunjukkan pada Gambar 4.48.



Edit Data Mata Kuliah

ID
49

Mata Kuliah
KI141302 Matematika Diskrit

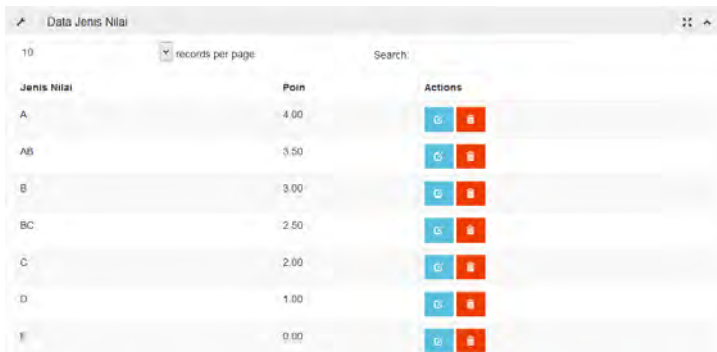
Rumpun Mata Kuliah
DTK (Dasar dan Terapan Koi)

Update **Batal**

Gambar 4.48 Form untuk Edit Data Mata Kuliah

c. Antarmuka Halaman Data Jenis Nilai

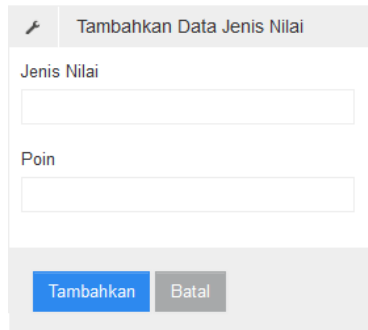
Antarmuka halaman data jenis nilai ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data jenis nilai pada *website*. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar jenis nilai dan bagian *form* menambah dan mengubah data jenis nilai. Data jenis nilai ini terdiri dari jenis nilai. Antarmuka halaman data jenis nilai dapat dilihat pada Gambar 4.49.



Jenis Nilai	Poin	Actions
A	4.00	
AB	3.50	
B	3.00	
BC	2.50	
C	2.00	
D	1.00	
E	0.00	

Gambar 4.49 Antarmuka Halaman Data Jenis Nilai

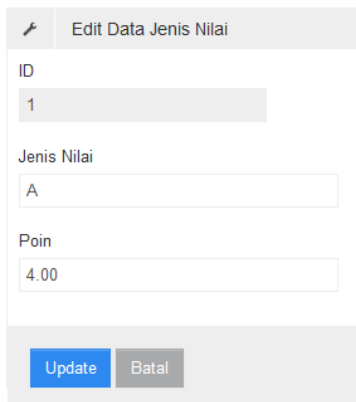
Jika ingin menambahkan data jenis nilai, maka admin master harus memasukkan nama jenis nilai dan poin. Form untuk menambahkan data mata kuliah ditunjukkan pada Gambar 4.50.



The form is titled 'Tambahkan Data Jenis Nilai' with a wrench icon. It contains two input fields: 'Jenis Nilai' and 'Poin'. At the bottom, there are two buttons: 'Tambahkan' (blue) and 'Batal' (grey).

Gambar 4.50 Form untuk Menambah Data Jenis Nilai

Pada halaman data jenis nilai, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data jenis nilai akan menampilkan *form* untuk *edit* data jenis nilai. Form edit ini ditunjukkan pada Gambar 4.51.



The form is titled 'Edit Data Jenis Nilai' with a wrench icon. It contains three input fields: 'ID' (with the value '1'), 'Jenis Nilai' (with the value 'A'), and 'Poin' (with the value '4.00'). At the bottom, there are two buttons: 'Update' (blue) and 'Batal' (grey).

Gambar 4.51 Form untuk Edit Data Jenis Nilai

d. Antarmuka Halaman Pemetaan

Antarmuka halaman pemetaan ini digunakan untuk melakukan pemetaan antara cabang kegiatan dengan rumpun mata kuliah. Halaman pemetaan menampilkan nama kegiatan, tahun kegiatan, dan cabang kegiatan yang dipetakan dengan rumpun mata kuliah. Admin dapat melakukan pemetaan dengan memilih *icon edit* pada kolom RMK. Setiap cabang kegiatan dapat dipetakan dengan banyak RMK. Antarmuka halaman pemetaan dapat dilihat pada Gambar 4.52.

Kegiatan	Tahun	Cabang	RMK
COMPFEST UI	2014	Intel® Real Sense™ Challenge	<ul style="list-style-type: none"> RPL (Rekayasa Perangkat Lunak)
COMPFEST UI	2014	Sumo Robot Battle	
COMPFEST UI	2014	Indie Game Ignite (GameDev)	
COMPFEST UI	2014	Competitive Programming	
COMPFEST UI	2014	Business IT Case	
GEMASTIK	2014	Penrograman	<ul style="list-style-type: none"> ALPRO (Algoritma dan Pemrograman)

Gambar 4.52 Antarmuka Halaman Pemetaan

Jika ingin melakukan pemetaan, maka admin master harus memilih *icon edit* pada kolom RMK. Kemudian sistem akan menampilkan halaman tambah pemetaan. Halaman tambah pemetaan terdiri dari tabel daftar RMK yang dipetakan dan *form* untuk menambah RMK baru untuk dipetakan. Halaman pemetaan dapat dilihat pada Gambar 4.53.

Kegiatan	Cabang	RMK	Action
COMPFEST UI	Intel® Real Sense™ Challenge	RPL (Rekayasa Perangkat Lunak)	

Gambar 4.53 Antarmuka Halaman Tambah Pemetaan

Tabel daftar RMK memuat nama kegiatan, cabang kegiatan, dan RMK. Tabel daftar RMK ditunjukkan pada Gambar 4.54.



Kegiatan	Cabang	RMK	Action
GEMASTIK	Pengembangan Game	RPL (Rekayasa Perangkat Lunak)	
GEMASTIK	Pengembangan Game	IGS (Interaksi, Grafika, dan Seni)	
GEMASTIK	Pengembangan Game	ALPRO (Algoritma dan Pemrograman)	

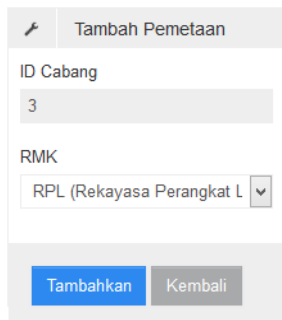
Showing 1 to 3 of 3 entries

Previous 1 Next

Kembali

Gambar 4.54 Tabel Daftar RMK

Untuk menambahkan RMK, maka admin dapat memasukan data baru pada *form* tambah pemetaan. *Form* tambah pemetaan diperlihatkan pada Gambar 4.55.



Tambah Pemetaan

ID Cabang

3

RMK

RPL (Rekayasa Perangkat Lunak)

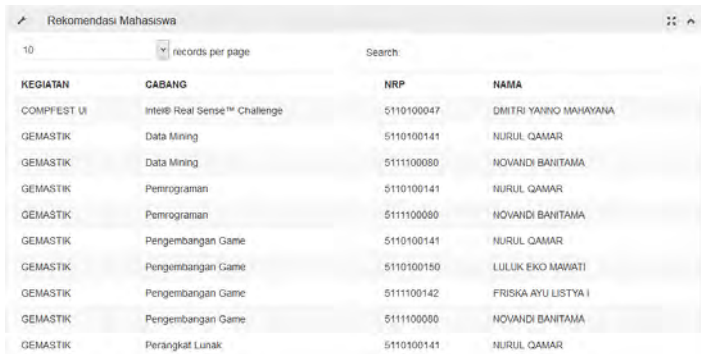
Tambahkan Kembali

Gambar 4.55 Form Tambah Pemetaan

e. Antarmuka Halaman Rekomendasi Mahasiswa

Antarmuka halaman rekomendasi mahasiswa menampilkan daftar nama mahasiswa yang direkomendasikan untuk mengikuti perlombaan pada satu cabang kegiatan.

Antarmuka halaman rekomendasi mahasiswa dapat dilihat pada Gambar 4.56.



Rekomendasi Mahasiswa

10 records per page Search:

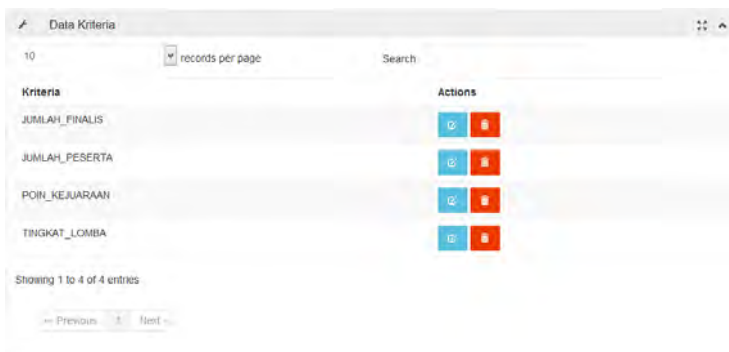
KEGIATAN	CABANG	NRP	NAMA
COMPETEST UI	Intel® Real Sense™ Challenge	5110100047	DMITRI YANNO MAHAYANA
GEMASTIK	Data Mining	5110100141	NURUL QAMAR
GEMASTIK	Data Mining	5111100080	NOVANDI BANITAMA
GEMASTIK	Pemrograman	5110100141	NURUL QAMAR
GEMASTIK	Pemrograman	5111100080	NOVANDI BANITAMA
GEMASTIK	Pengembangan Game	5110100141	NURUL QAMAR
GEMASTIK	Pengembangan Game	5110100150	LULUK EKO MAWATI
GEMASTIK	Pengembangan Game	5111100142	FRISKA AYU LISTYA I
GEMASTIK	Pengembangan Game	5111100080	NOVANDI BANITAMA
GEMASTIK	Perangkat Lunak	5110100141	NURUL QAMAR

Gambar 4.56 Antarmuka Halaman Rekomendasi Mahasiswa

4. Manajemen Proses SAW









a. Antarmuka Halaman Kriteria

Antarmuka halaman kriteria ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan kriteria. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar kriteria dan bagian *form* menambah dan mengubah kriteria. Antarmuka halaman dkriteria dapat dilihat pada Gambar 4.57.



Data Kriteria

10 records per page Search:

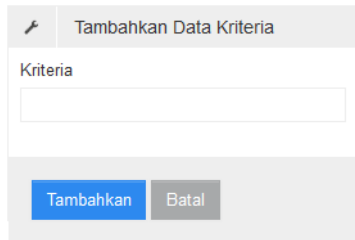
Kriteria	Actions
JUMLAH_FINALIS	 
JUMLAH_PESERTA	 
POIN_KELAJARAN	 
TINGKAT_LOMBA	 

Showing 1 to 4 of 4 entries

Previous Next

Gambar 4.57 Antarmuka Halaman Kriteria

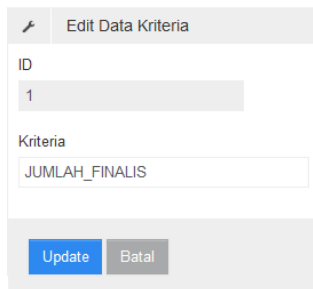
Jika ingin menambahkan kriteria, maka admin master harus memasukan nama kriteria. Form untuk menambahkan kriteria ditunjukan pada Gambar 4.58.



The image shows a web form titled 'Tambahkan Data Kriteria' (Add Criterion Data). It features a header bar with a wrench icon and the title. Below the header is a text input field labeled 'Kriteria'. At the bottom of the form are two buttons: 'Tambahkan' (Add) in blue and 'Batal' (Cancel) in grey.

Gambar 4.58 Form untuk Menambah Kriteria

Pada halaman kriteria, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman kriteria akan menampilkan *form* untuk *edit* kriteria. Form edit ini ditunjukan pada Gambar 4.59.



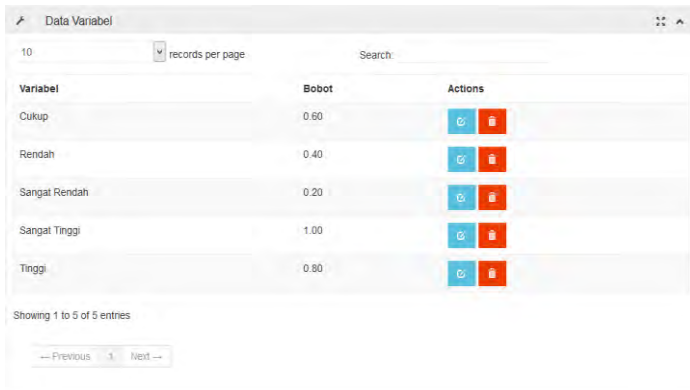
The image shows a web form titled 'Edit Data Kriteria' (Edit Criterion Data). It features a header bar with a wrench icon and the title. Below the header, there are two input fields: 'ID' with the value '1' and 'Kriteria' with the value 'JUMLAH_FINALIS'. At the bottom of the form are two buttons: 'Update' in blue and 'Batal' (Cancel) in grey.

Gambar 4.59 Form untuk Edit Kriteria











b. Antarmuka Halaman Variabel

Antarmuka halaman variabel ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan variabel. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar variabel dan bagian *form* menambah dan

mengubah variabel. Antarmuka halaman variabel dapat dilihat pada Gambar 4.60.

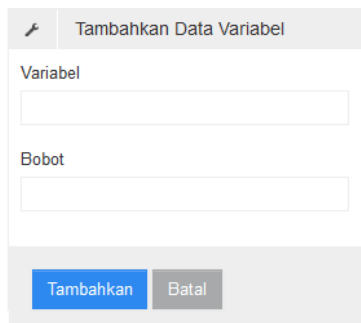


The screenshot shows a web interface titled "Data Variabel". At the top, there is a search bar and a dropdown menu set to "10 records per page". Below this is a table with three columns: "Variabel", "Bobot", and "Actions". The table contains five rows of data. Each row in the "Actions" column has two buttons: a blue "edit" icon and a red "delete" icon. At the bottom of the table, it says "Showing 1 to 5 of 5 entries" and has "Previous" and "Next" navigation links.

Variabel	Bobot	Actions
Cukup	0.60	 
Rendah	0.40	 
Sangat Rendah	0.20	 
Sangat Tinggi	1.00	 
Tinggi	0.80	 

Gambar 4.60 Antarmuka Halaman Variabel

Jika ingin menambahkan variabel, maka admin master harus memasukan nama variabel dan bobot. Form untuk menambahkan variabel ditunjukan pada Gambar 4.61.

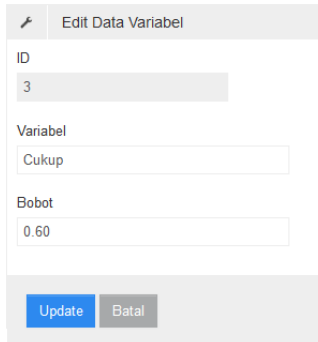


The screenshot shows a form titled "Tambahkan Data Variabel". It has two input fields: "Variabel" and "Bobot". At the bottom of the form, there are two buttons: "Tambahkan" (Add) in blue and "Batal" (Cancel) in gray.

Gambar 4.61 Form untuk Menambah Variabel

Pada halaman variabel, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman variabel akan menampilkan

form untuk *edit* variabel. Form edit ini ditunjukkan pada Gambar 4.62.

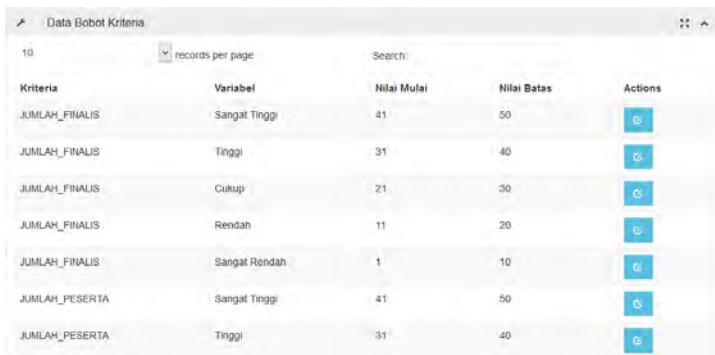


The image shows a web form titled "Edit Data Variabel". It contains three input fields: "ID" with the value "3", "Variabel" with the value "Cukup", and "Bobot" with the value "0.60". At the bottom, there are two buttons: "Update" (in blue) and "Batal" (in grey).

Gambar 4.62 Form untuk Edit Variabel

c. Antarmuka Halaman Atur Nilai Range

Antarmuka halaman atur nilai *range* ini digunakan untuk melakukan pengubahan nilai *range* dari setiap kriteria untuk setiap bobot variabel yang dimiliki. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar kriteria untuk setiap bobot variabelnya dan nilai *rangennya* dan bagian *form* untuk mengubah nilai *range*. Antarmuka halaman atur nilai *range* dapat dilihat pada Gambar 4.63.

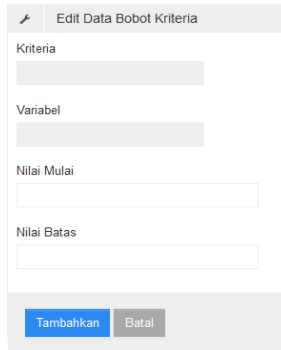


The image shows a web interface titled "Data Bobot Kriteria". It features a table with columns: "Kriteria", "Variabel", "Nilai Mulai", "Nilai Batas", and "Actions". The table contains seven rows of data. Above the table, there is a "records per page" dropdown set to "10" and a "Search:" input field. Each row in the table has a blue button with a minus sign in the "Actions" column.

Kriteria	Variabel	Nilai Mulai	Nilai Batas	Actions
JUMLAH_FINALIS	Sangat Tinggi	41	50	[Minus]
JUMLAH_FINALIS	Tinggi	31	40	[Minus]
JUMLAH_FINALIS	Cukup	21	30	[Minus]
JUMLAH_FINALIS	Rendah	11	20	[Minus]
JUMLAH_FINALIS	Sangat Rendah	1	10	[Minus]
JUMLAH_PESERTA	Sangat Tinggi	41	50	[Minus]
JUMLAH_PESERTA	Tinggi	31	40	[Minus]

Gambar 4.63 Antarmuka Halaman Atur Nilai Range

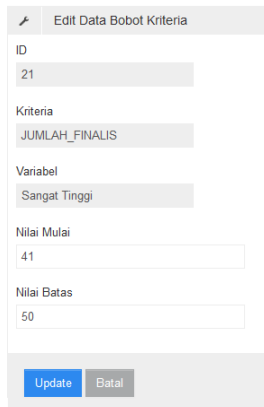
Jika ingin mengubah nilai *range*, maka admin master harus memilih nilai *range*. Form untuk mengubah nilai *range* ditunjukkan pada Gambar 4.64.



The form is titled "Edit Data Bobot Kriteria". It contains four input fields: "Kriteria", "Variabel", "Nilai Mulai", and "Nilai Batas". At the bottom, there are two buttons: "Tambahkan" (Add) in blue and "Batal" (Cancel) in grey.

Gambar 4.64 Form untuk Mengubah Nilai Range

Pada halaman atur nilai *range*, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman atur nilai *range* akan menampilkan *form* untuk *edit* nilai *range*. Form edit ini ditunjukkan pada Gambar 4.65.



The form is titled "Edit Data Bobot Kriteria". It contains five input fields: "ID" (with value 21), "Kriteria" (with value JUMLAH_FINALIS), "Variabel" (with value Sangat Tinggi), "Nilai Mulai" (with value 41), and "Nilai Batas" (with value 50). At the bottom, there are two buttons: "Update" in blue and "Batal" in grey.

Gambar 4.65 Form untuk Edit Nilai Range

d. Antarmuka Halaman Rekomendasi Prestasi

Antarmuka halaman rekomendasi prestasi menampilkan daftar rekomendasi prestasi yang dilengkapi tombol update, tanggal terakhir update data rekomendasi prestasi, dan label tanda data baru yang masuk dalam proses perhitungan rekomendasi prestasi. Tombol update digunakan untuk menampilkan rekomendasi berdasarkan data terbaru. Jika terdapat data baru masuk dalam proses perhitungan rekomendasi prestasi, hasil rekomendasi prestasi yang baru tidak langsung ditampilkan dalam website. Namun data tersebut disimpan dalam sebuah tabel, hal ini dilakukan untuk mengurangi beban kerja aplikasi. Tanggal terakhir update data rekomendasi prestasi digunakan untuk membantu admin master mengetahui waktu terakhir melakukan update data rekomendasi prestasi. Sedangkan label tanda data baru digunakan untuk membantu admin mengetahui jika terdapat data baru dalam proses perhitungan rekomendasi prestasi. Antarmuka halaman rekomendasi prestasi dapat dilihat pada Gambar 4.66.

Ranking	TAHUN	KEGIATAN	CABANG	PESERTA	FINALIS	POIN JUARA	POIN TINGKAT	Value
1	2014	GEMASTIK	Data Mining	0.25	1.00	1.00	1.00	2.45
2	2014	GEMASTIK	Perangkat Lunak	0.25	1.00	0.00	1.00	2.25
3	2014	GEMASTIK	Keamanan Jaringan	1.00	1.00	0.00	1.00	1.00
4	2014	GEMASTIK	Pernikahan Bisnis	0.25	1.00	0.00	1.00	1.45
5	2014	GEMASTIK	Pengembangan Game	0.50	0.00	0.00	1.00	0.70
6	2014	GEMASTIK	Penrograman	0.25	0.00	0.00	1.00	0.65
7	2014	GEMASTIK	Piranti Cerdas	0.25	0.00	0.00	1.00	0.65
8	2014	INAICTA	Application : Perguruan Tinggi	0.25	0.00	0.00	1.00	0.65

Showing 1 to 8 of 8 entries

— Previous 1 Next —

Data terbaru Terakhir Update : 12/23/2014 4:39:50 PM

[Update](#)

Gambar 4.66 Antarmuka Halaman Rekomendasi Prestasi

5. Manajemen Status


a. Antarmuka Halaman Data Status Prestasi

Antarmuka halaman data status prestasi ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data status prestasi. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar data status prestasi dan bagian *form* menambah dan mengubah data status prestasi. Antarmuka halaman data status prestasi dapat dilihat pada Gambar 4.67.

Status Prestasi	Actions
Finalis	 
Juara	 
Penyisihan	 
Peserta	 

Gambar 4.67 Antarmuka Halaman Data Status Prestasi

Jika ingin menambahkan data status prestasi, maka admin master harus memasukan nama status prestasi. Form untuk menambahkan data status prestasi ditunjukkan pada Gambar 4.68.


 Tambahkan Data Status Prestasi

Status Prestasi

Tambahkan
 Batal

Gambar 4.68 Form untuk Menambah Data Status Prestasi

Pada halaman data status prestasi, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data status prestasi akan menampilkan *form* untuk *edit* data status prestasi. Form edit ini ditunjukkan pada Gambar 4.69.

Edit Data Status Prestasi

ID

3

Status Prestasi

Finalis

Update

Batal

Gambar 4.69 Form untuk Edit Data Status Prestasi

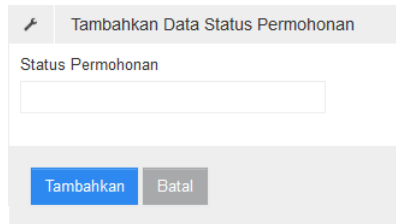
b. Antarmuka Halaman Data Status Permohonan

Antarmuka halaman data status permohonan ini digunakan untuk melakukan penambahan, perubahan, dan penghapusan data status permohonan. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar data status permohonan dan bagian *form* menambah dan mengubah data status prestasi. Antarmuka halaman data status permohonan dapat dilihat pada Gambar 4.70.

Status Permohonan	Actions
<div>Ditolak</div>	<div></div> <div></div>
<div>Mengajukan</div>	<div></div> <div></div>
<div>OK</div>	<div></div> <div></div>
<div>Revisi</div>	<div></div> <div></div>

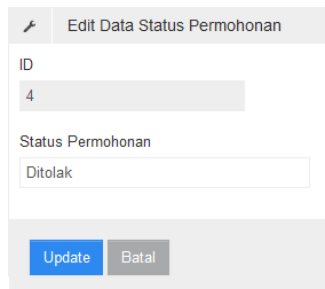
Gambar 4.70 Antarmuka Halaman Data Status Permohonan

Jika ingin menambahkan data status permohonan, maka admin master harus memasukan nama status permohonan. Form untuk menambahkan data status permohonan ditunjukan pada Gambar 4.71.



Gambar 4.71 Form untuk Menambah Data Status Permohonan





Pada halaman data status permohonan, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data status permohonan akan menampilkan *form* untuk *edit* data status permohonan. Form edit ini ditunjukkan pada Gambar 4.72.



Gambar 4.72 Form untuk Edit Data Status Permohonan


c. Antarmuka Halaman Data Status Mahasiswa

Antarmuka halaman data status mahasiswa ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data status mahasiswa. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar data status mahasiswa dan bagian *form* menambah dan mengubah data status mahasiswa. Antarmuka halaman data status mahasiswa dapat dilihat pada Gambar 4.73.

Status Mahasiswa	Actions
Lulus	 
Mahasiswa	 

Gambar 4.73 Antarmuka Halaman Data Status Mahasiswa

Jika ingin menambahkan data status mahasiswa, maka admin master harus memasukan nama status mahasiswa. Form untuk menambahkan data status mahasiswa ditunjukan pada Gambar 4.74.

 Tambahkan Data Status Mahasiswa

Status Mahasiswa

Tambahkan

Batal

Gambar 4.74 Form untuk Menambah Data Status Mahasiswa

Pada halaman data status mahasiswa, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data status mahasiswa akan menampilkan *form* untuk *edit* data status mahasiswa. Form edit ini ditunjukan pada Gambar 4.75.

✎ Edit Data Status Mahasiswa

ID
1





Status Mahasiswa
Lulus

Update Batal

Gambar 4.75 Form untuk Edit Data Status Mahasiswa

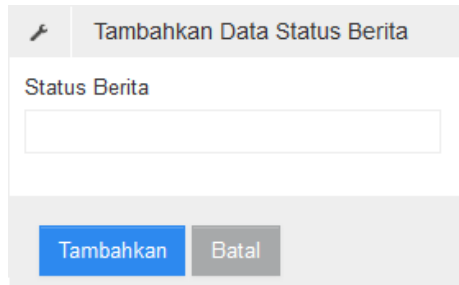
d. Antarmuka Halaman Data Status Berita

Antarmuka halaman data status berita ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan data status berita. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar data status berita dan bagian *form* menambah dan mengubah data status berita. Antarmuka halaman data status berita dapat dilihat pada Gambar 4.76.

Status Berita	Actions
Draft	 
Final	 

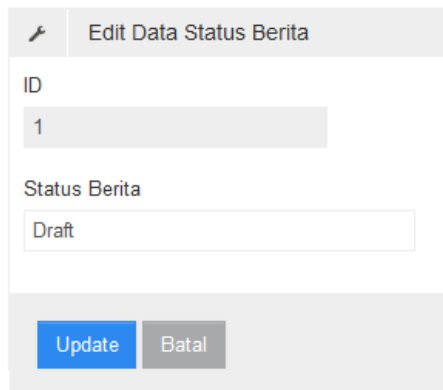
Gambar 4.76 Antarmuka Halaman Data Status Berita

Jika ingin menambahkan data status berita, maka admin master harus memasukan nama status berita. Form untuk menambahkan data status berita ditunjukan pada Gambar 4.77.



Gambar 4.77 Form untuk Menambah Data Status Berita

Pada halaman data status berita, admin dapat melakukan perubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data status berita akan menampilkan *form* untuk *edit* data status berita. Form edit ini ditunjukkan pada Gambar 4.78.



Gambar 4.78 Form untuk Edit Data Status Berita

e. Antarmuka Halaman Data *Role*


Antarmuka halaman data *role* ini digunakan untuk melakukan penambahan, perubahan, dan penghapusan data status prestasi. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar data *role* dan bagian *form* menambah

dan mengubah data *role*. Antarmuka halaman data *role* dapat dilihat pada Gambar 4.79.

Nama Role	Actions
AdminBerita	 
AdminMaster	 
AdminPermohonan	 
Mahasiswa	 

Gambar 4.79 Antarmuka Halaman Data Role

Jika ingin menambahkan data *role*, maka admin master harus memasukan nama *role*. Form untuk menambahkan data *role* ditunjukkan pada Gambar 4.80.


Tambahkan Data Role

Nama Role

Tambahkan
Batal

Gambar 4.80 Form untuk Menambah Data Role

Pada halaman data *role*, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman data *role* akan menampilkan *form* untuk *edit* data *role*. Form edit ini ditunjukkan pada Gambar 4.81.

Edit Data Status Prestasi

ID

3

Status Prestasi

Finalis

Update

Batal

Gambar 4.81 Form untuk Edit Data *Role*

6. Manajemen Berita

Antarmuka halaman manajemen berita ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan berita. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar berita dan bagian *form* menambah dan mengubah berita. Antarmuka halaman manajemen berita dapat dilihat pada Gambar 4.82.

Judul	Berita	Tanggal	Gambar	Status	Actions
asadasd	adasdasdad	12/18/2014 3:11:16 PM		Draft	
Informatika ITS Juara Imagine Cup 2014	Mahasiswa Teknik Informatika ITS menjuarai kompetisi Imagine Cup 2014 kategori World Citizenship. Tim yang beranggotakan Muhammad Rizky Habibi, Nurul Wakhidatul Ummah, Mentari Queen Glossyta, dan Fiandra Fatarany mempresentasikan aplikasi bernama "Cakra" yaitu aplikasi terapi autisme yang mereka kembangkan dan berhasil memukau penonton serta dewan juri yang memenuhi Epicentrum Vaik, Jakarta. Dengan prestasi ini, mereka berhak untuk mewakili Indonesia di ajang World Semifinal.	12/21/2014 4:29:27 PM		Final	
Mahasiswa Teknik Informatika juara LCEN XVIII 2014 Kategori Medical Electronics & Assistive Technology	Mahasiswa Teknik Informatika kembali lagi meraih prestasi. Kali ini, tim Digid2 yang dimotori oleh Muhammad Rizky Habibi, Nurul Wakhidatul Ummah, dkk. berhasil menjadi "1st Winner Medical Electronics & Assistive Technology" pada LCEN XVIII 2014 (Lomba Cipta Elektroteknik Nasional) yang diselenggarakan oleh Jurusan Teknik Elektro, ITS di Tunjungan Plaza Convention Hall, 24-25 Mei 2014. Selamat kepada tim Digid2 atas prestasi yang telah diraih dan semoga semakin sukses. Informasi lebih lanjut mengenai produk yang dibuat oleh tim Digid2 dapat dilihat pada https://www.facebook.com/itakra.digid2 . (mm)	11/19/2014 1:29:19 PM		Final	

Gambar 4.82 Antarmuka Halaman Manajemen Berita

Jika ingin menambahkan berita, maka admin master harus mengunggah gambar, judul berita, dan isi berita. Form untuk menambahkan berita ditunjukkan pada Gambar 4.83.

Pada halaman manajemen berita, admin berita dan admin master dapat melakukan perubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman berita akan menampilkan *form* untuk *edit* data *role*. Form edit ini ditunjukkan pada Gambar 4.84.

Tambahkan Berita

Gambar

No file selected Choose File

Judul

Isi

Status

Draft

Tambahkan Batal

Gambar 4.83 Form untuk Menambah Berita

Edit Berita

ID
1

Tanggal
11/19/2014 1:29:19 PM

Gamber
No file select 201412081206_Mahasiswa-Teknik-Informatika-juara-LCEN-XVII-2014-Kategori-Medical-Electronics-&-Assistive-Technology.jpg

Judul
Mahasiswa Teknik Informatika Juara LCEN XV

Isi
 B U X T H T [Rich Text Editor]
 Mahasiswa Teknik Informatika kembali lagi meraih prestasi. Kali ini, tim Digid2 yang didanai oleh Muhammad Rizky Habbil, Nurul Wakhidatul Ummah, dan berhasil menjadi "1st Winner Medical Electronics & Assistive Technology" pada LCEN XVII 2014 (Lomba Cipta Elektronika Nasional) yang diselenggarakan oleh Jurusan Teknik Elektro, ITS di Tunjungan Plaza Convention Hall, 24-25 Mei 2014. Selamat kepada tim Digid2 atas prestasi yang telah diraih dan semoga semakin sukses.
 Informasi lebih lanjut mengenai produk yang dibuat oleh tim Digid2 dapat dilihat pada <https://www.facebook.com/cakra.digid2> (mp)

Status
Final

[Update] [Batal]

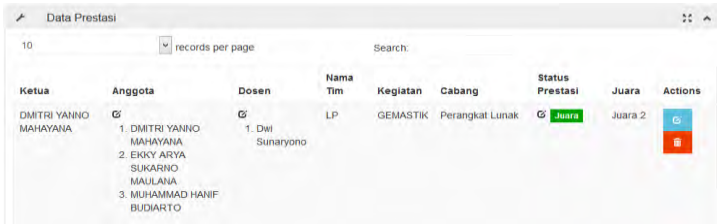
Gambar 4.84 Form untuk Edit Berita


4.2.2.4. Implementasi Antarmuka Mahasiswa

1. Info Prestasi

Antarmuka halaman info prestasi ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan info prestasi mahasiswa. Info prestasi terdiri dari nama ketua, nama anggota tim, nama dosen pembimbing, nama tim, nama kegiatan, nama cabang kegiatan, tahun kegiatan, status prestasi, dan tingkat juara. Pada kolom anggota, dosen pembimbing, status prestasi, dan juara terdapat *icon* untuk melakukan *edit*. Dimana mahasiswa dapat menambah nama anggota sejumlah jumlah anggota yang dimasukkan. Nama dosen pembimbing dapat ditambahkan sesuai dengan kebutuhan. Status prestasi dapat diubah jika info prestasi mengalami perubahan tahap,

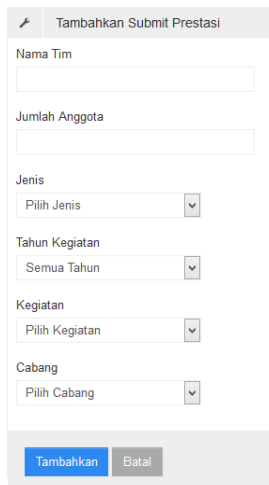
misalkan dari tahap peserta menjadi tahap penyisihan. Untuk juara sedikit berbeda dengan lainnya, *icon edit* akan muncul pada kolom juara jika status prestasi telah berubah menjadi juara. Antarmuka halaman info prestasi diperlihatkan pada Gambar 4.85.



Ketua	Anggota	Dosen	Nama Tim	Kegiatan	Cabang	Status Prestasi	Juara	Actions
DMITRI YANNO MAHAYANA	1. DMITRI YANNO MAHAYANA 2. EIKY ARYA SUKARNIO MAJULANA 3. MUHAMMAD HANIF BUDIARTO	1. Dwi Sunaryono	LP	GEMASTIK	Perangkat Lunak	Juara	Juara 2	 

Gambar 4.85 Antarmuka Halaman Info Prestasi

Jika ingin menambahkan info prestasi, maka mahasiswa harus memilih NRP ketua, memasukkan nama tim, jumlah anggota, memilih jenis kegiatan, tahun kegiatan, dan cabang kegiatan. Form untuk menambahkan info prestasi ditunjukkan pada Gambar 4.86.



Tambahkan Submit Prestasi

Nama Tim

Jumlah Anggota

Jenis

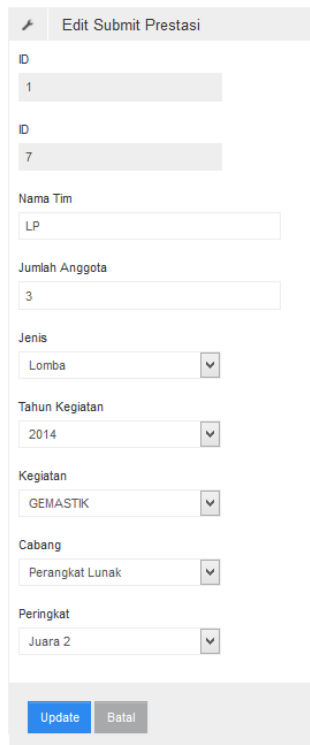
Tahun Kegiatan

Kegiatan

Cabang

Gambar 4.86 Form untuk Menambah Info Prestasi

Pada halaman info prestasi, admin dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman info prestasi akan menampilkan *form* untuk *edit* info prestasi. Form edit ini ditunjukkan pada Gambar 4.87.



Edit Submit Prestasi

ID
1

ID
7

Nama Tim
LP

Jumlah Anggota
3

Jenis
Lomba

Tahun Kegiatan
2014

Kegiatan
GEMASTIK

Cabang
Perangkat Lunak

Peringkat
Juara 2

Update **Batal**

Gambar 4.87 Form untuk Edit Info Prestasi


Data anggota pada info prestasi dapat diperbarui dengan melakukan klik pada *icon edit* yang terletak dalam kolom anggota. Kemudian akan muncul antarmuka halaman tambah anggota yang terbagi menjadi dua bagian yaitu tabel daftar anggota dan *form* untuk menambahkan anggota. Gambar

4.88 merupakan halaman tambah anggota yang menampilkan data anggota yang terdiri dari NRP dan nama.

NRP	Nama	Action
5110100047	DMITRI YANNO MAHAYANA	
5110100087	EKKY ARYA SUKARNO MAULANA	
5110100202	MUHAMMAD HANIF BUDIARTO	

Gambar 4.88 Antarmuka Halaman Mahasiswa Tambah Anggota

Jika mahasiswa ingin menambahkan anggota maka mahasiswa hanya perlu memilih nama mahasiswa yang telah tersimpan dalam sistem. Form untuk tambah anggota dapat dilihat pada Gambar 4.89.

 Tambah Anggota

ID Prestasi

Nama Mahasiswa


Gambar 4.89 Form untuk Tambah Anggota

Data dosen pada info prestasi dapat diperbarui dengan melakukan klik pada *icon edit* yang terletak dalam kolom dosen. Kemudian akan muncul antarmuka halaman tambah dosen yang terbagi menjadi dua bagian yaitu tabel daftar dosen dan *form* untuk menambahkan dosen. Gambar 4.90 merupakan halaman tambah dosen yang menampilkan data anggota yang terdiri dari NIP dan nama.

NIP	Nama	Action
197205281997021000	Dwi Sunaryono	

Gambar 4.90 Antarmuka Halaman Mahasiswa Tambah Dosen

Jika mahasiswa ingin menambahkan dosen maka mahasiswa hanya perlu memilih nama dosen yang telah tersimpan dalam sistem. Form untuk tambah dosen dapat dilihat pada Gambar 4.91.

 Tambah Dosen

ID Prestasi

1

Nama Dosen

Supeno Djanali

Tambahkan

Kembali

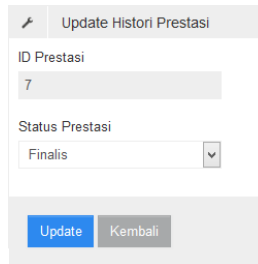
Gambar 4.91 Form untuk Tambah Dosen

Status prestasi pada info prestasi dapat diperbarui dengan melakukan klik pada *icon edit* yang terletak dalam kolom status prestasi. Kemudian akan muncul antarmuka halaman histori prestasi untuk *update* status prestasi yang terbagi menjadi dua bagian yaitu tabel daftar histori status prestasi dan *form* untuk memperbarui status prestasi. Gambar 4.92 merupakan halaman *update* histori prestasi yang menampilkan daftar prestasi yang terdiri dari nama tim, tanggal *update* status prestasi, dan status prestasi.

Prestasi	Tanggal	Status Prestasi	Actions
LP	9/4/2014 12:00:00 AM	Peserta	
LP	9/25/2014 12:00:00 AM	Penyisihan	
LP	10/23/2014 12:00:00 AM	Finalis	
LP	11/17/2014 5:04:41 PM	Juara	

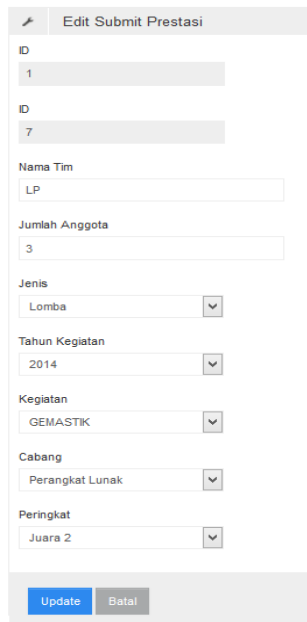
Gambar 4.92 Antarmuka Halaman Histori Prestasi

Jika mahasiswa ingin memperbarui status prestasi maka mahasiswa hanya perlu memilih status prestasi yang telah tersimpan dalam sistem. Form untuk tambah anggota dapat dilihat pada Gambar 4.93.



The screenshot shows a web form titled "Update Histori Prestasi". It contains two input fields: "ID Prestasi" with the value "7" and "Status Prestasi" with a dropdown menu showing "Finalis". At the bottom, there are two buttons: "Update" (blue) and "Kembali" (grey).

Gambar 4.93 Form untuk *Update* Status Prestasi





The screenshot shows a web form titled "Edit Submit Prestasi". It contains several input fields and dropdown menus: "ID" (1), "ID" (7), "Nama Tim" (LP), "Jumlah Anggota" (3), "Jenis" (Lomba), "Tahun Kegiatan" (2014), "Kegiatan" (GEMASTIK), "Cabang" (Perangkat Lunak), and "Peringkat" (Juara 2). At the bottom, there are two buttons: "Update" (blue) and "Batal" (grey).

Gambar 4.94 Antarmuka Halaman Form Edit Info Prestasi dengan Peringkat

Juara pada info prestasi dapat diperbarui jika status prestasi tim telah menjadi juara. Juara ini dapat diperbarui dengan melakukan klik pada *icon edit* yang terletak dalam kolom *Actions*. Kemudian akan muncul *form edit* info prestasi dengan tambahan *dropdown* untuk *update* peringkat. Gambar 4.94 menunjukkan *form edit* info prestasi dengan tambahan *dropdown update* peringkat.

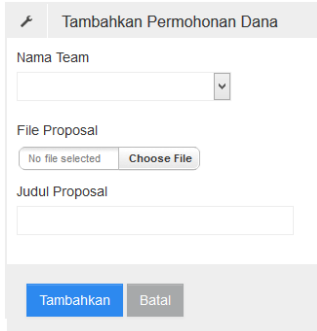
2. **Permohonan Dana**

Antarmuka halaman permohonan dana ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan permohonan dana. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar permohonan dana dan bagian *form* menambah dan mengubah permohonan dana. Permohonan dana terdiri dari nama ketua, nama tim, nama kegiatan, cabang kegiatan, judul proposal, *file* proposal, status prestasi, dan status permohonan dana. Antarmuka halaman permohonan dana diperlihatkan pada Gambar 4.95.

Ketua	Nama Tim	Kegiatan	Cabang	Judul	File	Status Prestasi	Status Permohonan	Actions
DMITRI YANNO MAHAYANA	LP	GEMASTIK	Perangkat Lunak	LPJuara	5110100047_1_201412212052.pdf	Juara	Mengajukan	 

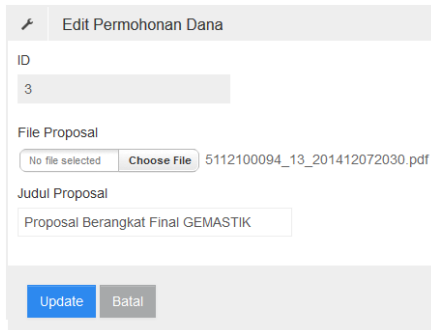
Gambar 4.95 Antarmuka Halaman Permohonan Dana

Jika ingin menambahkan permohonan dana, maka mahasiswa harus memilih nama tim, mengunggah *file* proposal, dan memasukan judul proposal. Bentuk *file* proposal yang dapat diunggah adalah *file* dalam bentuk *pdf* dan *word*. Sedangkan untuk status permohonan *default* mengajukan, status dari status permohonan dana akan merubah jika sudah diubah oleh admin yang bertanggungjawab. Form untuk menambahkan permohonan dana ditunjukan pada Gambar 4.96.



Gambar 4.96 Form untuk Menambah Permohonan Dana

Pada halaman permohonan dana, mahasiswa dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman permohonan dana akan menampilkan *form* untuk *edit* permohonan dana. Form edit ini ditunjukkan pada Gambar 4.97.



Gambar 4.97 Form untuk Edit Permohonan Dana

3. Nilai Kuliah


Antarmuka halaman nilai kuliah ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan nilai kuliah. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar nilai kuliah dan bagian *form* menambah dan mengubah nilai kuliah. Nilai kuliah terdiri dari nama RMK,

nama mata kuliah, dan jenis nilai. Antarmuka halaman nilai kuliah diperlihatkan pada Gambar 4.98.

RMK	Mata Kuliah	Nilai	Actions
RPL (Rekayasa Perangkat Lunak)	K141325 Perancangan Perangkat Lunak	AB	 

Gambar 4.98 Antarmuka Halaman Nilai Kuliah Mahasiswa

Jika ingin menambahkan nilai kuliah, maka mahasiswa harus memilih RMK, memilih mata kuliah, dan memilih jenis nilai. Form untuk menambahkan nilai kuliah ditunjukkan pada Gambar 4.99.



Tambahkan Submit Nilai

RMK

Pilih RMK

▼

Mata Kuliah

Pilih Makul

▼

Nilai

A

▼

Tambahkan

Batal

Gambar 4.99 Form untuk Menambah Nilai Kuliah Mahasiswa

Pada halaman nilai kuliah, mahasiswa dapat melakukan pengubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*. Kemudian halaman nilai kuliah akan menampilkan *form edit* nilai kuliah. Form edit ini ditunjukkan pada Gambar 4.100.

Edit Submit Nilai

ID
5

NRP
5110100150

RMK
IGS (Interaksi, Grafika, dan S)

Mata Kuliah
KI141402 Animasi Komputer

Nilai
A

Update **Batal**

Gambar 4.100 Form untuk Edit Nilai Kuliah Mahasiswa

4. Berita

Antarmuka halaman berita ini digunakan untuk melakukan penambahan, pengubahan, dan penghapusan berita. Halaman ini terbagi menjadi dua bagian yaitu bagian untuk tabel daftar berita dan bagian *form* menambah dan mengubah berita. Antarmuka halaman berita dapat dilihat pada Gambar 4.101.

Judul	Berita	Tanggal	Gambar	Status	Actions
No data available in table					

Gambar 4.101 Antarmuka Halaman Berita

Jika ingin menambahkan berita, maka mahasiswa harus mengunggah gambar, judul berita, dan isi berita. Form untuk menambahkan berita ditunjukkan pada Gambar 4.102.

Gambar 4.102 Form untuk Menambah Berita

Pada halaman manajemen berita, mahasiswa dapat melakukan perubahan data dengan memilih *icon edit* berwarna biru pada kolom *actions*.

4.2.3. Implementasi Control

Implementasi yang ada pada lapisan kontrol ini merupakan kelas-kelas yang bertanggung jawab atas perilaku sistem. Kelas-kelas pada lapisan ini digunakan untuk menghubungkan kelas pada lapisan antarmuka dengan kelas yang ada pada lapisan data.

4.2.3.1. Kelas DosenController

Kelas controller ini digunakan untuk memuat kelas yang mengatur proses bisnis yang ditangani oleh admin. *DosenController* ini terbagi menjadi 29 *region* yang

didalamnya terdapat *class controller* untuk *view*. Untuk bagian *controller* rekomendasi mahasiswa terdapat *sub class* dan *query*, sedangkan untuk *controller* rekomendasi prestasi terdiri dari *query* dan fungsi *typebuilder*. 29 region yang terdapat pada *class controller* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Daftar Region dalam Kelas DosenController

No	Region	No	Region
1	Entities	16	Data Status Berita
2	Index	17	Data Status Permohonan
3	Profil	18	Data Status Prestasi
4	Data Pengguna	19	Data Role
5	Data Mahasiswa	20	Kriteria
6	Data Dosen	21	Variabel
7	Data Peringkat	22	Data Prestasi
8	Data Jenis Kegiatan	23	Laporan Prestasi
9	Data Tingkat Kegiatan	24	Permohonan Dana
10	Data Kegiatan	25	Pemetaan
11	Data Cabang Kegiatan	26	Rekomendasi Mahasiswa
12	Data Bidang	27	Berita
13	Data Mata Kuliah	28	Bobot Kriteria
14	Data Jenis Nilai	29	Rekomendasi Prestasi
15	Data Status Mahasiswa	30	-

Pada region entities terdapat kode untuk mengakses database. Kode untuk entitas ditunjukkan pada Kode Sumber 4.2

```
#region Entities
private Entities db = new Entities();
protected override void Dispose(bool disposing)
{
    db.Dispose();
    base.Dispose(disposing);
}
#endregion
```

Kode Sumber 4.2 Region Entities

Pada region index terdapat *action method* Index() yang berisi perintah untuk menampilkan rekomendasi prestasi dan data prestasi terbaru. Kemudian terdapat pula *action method* UpdateRekomendasiPrestasiIndex() yang berfungsi untuk melakukan perhitungan proses *Simple Additive Weighting*. Proses perhitungan dari *Simple Additive Weighting* melalui beberapa tahap yaitu menentukan alternatif, menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, menentukan rating kecocokan setiap alternatif pada setiap kriteria yang telah ditentukan dan membuat matriks keputusan berdasarkan kriteria. Proses tersebut ditunjukkan pada Kode Sumber 4.3.

```
SELECT TA.TAHUN_KEGIATAN, TA.NAMA_KEGIATAN, TA.NAMA_CABANG,
SUM(TA.PESERTA) AS PESERTA, SUM(TA.FINALIS) AS FINALIS,
SUM(TA.POIN_JUARA) AS POIN_JUARA, SUM(TA.POIN_TINGKAT) AS
POIN_TINGKAT FROM
(select P.NAMA_KEGIATAN, P.NAMA_CABANG, P.TAHUN_KEGIATAN,
PESERTA=CASE N.ID_KRITERIA
WHEN 1 THEN
(
CASE
WHEN P.peserta BETWEEN N.NILAI_MULAI AND N.NILAI_BATAS
THEN
N.BOBOT_VARIABEL
ELSE 0
END
)
ELSE 0
END,
FINALIS=CASE N.ID_KRITERIA
WHEN 2 THEN
(
CASE
WHEN P.finalis BETWEEN N.NILAI_MULAI AND N.NILAI_BATAS
THEN
N.BOBOT_VARIABEL
ELSE 0
END
)
ELSE 0
END,
```

```

POIN_JUARA =CASE N.ID_KRITERIA
WHEN 3 THEN
(
CASE
WHEN P.poin_juara BETWEEN N.NILAI_MULAI AND
N.NILAI_BATAS THEN
N.BOBOT_VARIABEL
ELSE 0
END
)
ELSE 0
END,

POIN_TINGKAT=CASE N.ID_KRITERIA
WHEN 4 THEN
(
CASE
WHEN P.poin_tingkat BETWEEN N.NILAI_MULAI AND
N.NILAI_BATAS THEN
N.BOBOT_VARIABEL
ELSE 0
END
)
ELSE 0
END
from (

select a.TAHUN_KEGIATAN, a.NAMA_KEGIATAN,a.NAMA_CABANG,
SUM(a.PESERTA) as peserta, SUM(a.FINALIS) as finalis,
SUM(a.POIN_JUARA) as poin_juara, a.POIN_TINGKAT as
poin_tingkat from (
SELECT
KEGIATAN_PRESTASI.TAHUN_KEGIATAN,KEGIATAN_PRESTASI.NAMA_KEGIAT
AN,CABANG_KEGIATAN.NAMA_CABANG,

                                PESERTA=CASE
STATUS_PRESTASI.NAMA_STATUS_PRESTASI
                                WHEN 'PESERTA' THEN
SUM(PRESTASI.JUMLAH_ANGGOTA)
                                ELSE 0
                                END,

                                FINALIS=CASE
STATUS_PRESTASI.NAMA_STATUS_PRESTASI

```

```

                                WHEN 'FINALIS' THEN
SUM(PRESTASI.JUMLAH_ANGGOTA)
                                ELSE 0
                                END,

                                POIN_JUARA=CASE
STATUS_PRESTASI.NAMA_STATUS_PRESTASI
                                WHEN 'JUARA' THEN
SUM(PERINGKAT_JUARA.POIN_JUARA)
                                ELSE 0
                                END,

                                TINGKAT_KEGIATAN.POIN_TINGKAT

                                from
                                PRESTASI
                                LEFT JOIN HISTORY_PRESTASI ON
(PRESTASI.ID_PRESTASI=HISTORY_PRESTASI.ID_PRESTASI)
                                LEFT JOIN STATUS_PRESTASI ON
(HISTORY_PRESTASI.ID_STATUS_PRESTASI=STATUS_PRESTASI.ID_STATUS
_PRESTASI)
                                LEFT JOIN CABANG_KEGIATAN ON
(PRESTASI.ID_CABANG=CABANG_KEGIATAN.ID_CABANG)
                                LEFT JOIN KEGIATAN_PRESTASI ON
(CABANG_KEGIATAN.ID_KEGIATAN_PRESTASI=KEGIATAN_PRESTASI.ID_KEG
IATAN_PRESTASI)
                                LEFT JOIN PERINGKAT_JUARA ON
(PRESTASI.ID_PERINGKAT_JUARA=PERINGKAT_JUARA.ID_PERINGKAT_JUAR
A)
                                LEFT JOIN TINGKAT_KEGIATAN ON
(KEGIATAN_PRESTASI.ID_TINGKAT_KEGIATAN=TINGKAT_KEGIATAN.ID_TIN
GKAT_KEGIATAN)
group by KEGIATAN_PRESTASI.NAMA_KEGIATAN,
CABANG_KEGIATAN.NAMA_CABANG, KEGIATAN_PRESTASI.TAHUN_KEGIATAN,
STATUS_PRESTASI.NAMA_STATUS_PRESTASI,
TINGKAT_KEGIATAN.POIN_TINGKAT
) as a
group by a.NAMA_KEGIATAN, a.NAMA_CABANG, a.TAHUN_KEGIATAN,
a.POIN_TINGKAT
)P,
(SELECT NILAI_RANGE.ID_KRITERIA, NILAI_RANGE.ID_VARIABEL,
KRITERIA.NAMA_KRITERIA, NILAI_RANGE.NILAI_MULAI,
NILAI_RANGE.NILAI_BATAS, VARIABEL.NAMA_VARIABEL,
VARIABEL.BOBOT_VARIABEL FROM KRITERIA INNER JOIN NILAI_RANGE
ON KRITERIA.ID_KRITERIA = NILAI_RANGE.ID_KRITERIA INNER JOIN
VARIABEL ON NILAI_RANGE.ID_VARIABEL = VARIABEL.ID_VARIABEL)
N

```

) TA
GROUP BY TA.NAMA_KEGIATAN, TA.NAMA_CABANG, TA.TAHUN_KEGIATAN

Kode Sumber 4.3 *Query* untuk Bobot untuk Setiap Kriteria

Kode Sumber 4.3 merupakan matriks keputusan berdasarkan kriteria. Contoh dari hasil *query* tersebut ditunjukkan pada Gambar 4.103.

	TAHUN_KEGIATAN	NAMA_KEGIATAN	NAMA_CABANG	PESERTA	FINALIS	POIN_JUARA	POIN_TINGKAT
1	2013	COMPFEST UI	Competitive Programming	0.20	0.00	0.00	1.00
2	2014	GEMASTIK	Keamanan Jaringan	0.80	0.20	0.00	0.80
3	2014	GEMASTIK	Pengembangan Game	0.40	0.00	0.00	0.80
4	2014	GEMASTIK	Perangkat Lunak	0.20	0.20	0.80	0.80
5	2014	GEMASTIK	Piranti Cerdas	0.20	0.00	0.00	0.80
6	2014	INACTA	Application : Perguruan Tinggi	0.20	0.00	0.00	0.80
7	2014	PKM	PKM – Karya Cipta (PKM KC)	0.20	0.00	0.00	0.80

Gambar 4.103 Contoh Hasil Query dari Data Sementara

Selanjutnya dari matriks yang telah terbentuk dari hasil query dilakukanlah normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R . Menentukan nilai profesi yang diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan bobot ranking sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi. Proses tersebut dapat dilihat pada *action method* `UpdateRekomendasiPrestasiIndex()` yang ditunjukkan pada Kode Sumber 9.1 Lampiran B.

Pada region profil terdapat `ActionResult` dengan tipe `ViewResult` yang mengembalikan nilai dalam bentuk `View`. `ViewResult` profil berfungsi untuk menampilkan data profil pengguna serta terdapat `[HttpPost]` `ViewResult` Profil yang berfungsi untuk menyimpan data baru yang baru jika admin melakukan perubahan data serta mengembalikan dalam sebuah `View`. Kode sumber untuk region profil dapat dilihat pada Kode Sumber 9.2 Lampiran B.

Dalam `DosenController` terdapat beberapa region yang mempunyai fungsi method yang sama yaitu Data Pengguna, Data Mahasiswa, Data Dosen, Data Peringkat, Data Jenis Kegiatan, Data Tingkat Kegiatan, Data Kegiatan, Data Cabang Kegiatan, Data Bidang, Data Mata Kuliah, Data Jenis Nilai, Data Status Berita, Data Status Permohonan, Data Status Prestasi, Data Role, Kriteria, dan Variabel. Semua region mempunyai *action method* `ActionResult` yang mengembalikan dalam bentuk `View` dan *action method* `ActionResult` untuk melakukan Create, Edit, dan Delete. Kode Sumber 9.3 menunjukkan kode dari region Data Pengguna, didalamnya terdapat `ActionResult Create`, `ActionResult Edit`, dan `ActionResult DeletePengguna` yang ditunjukkan pada Kode Sumber 4.4, Kode Sumber 4.5, dan Kode Sumber 4.6.

```
[ActionName("DataPengguna")]
[OnAction(ButtonName = "Create")]
public ActionResult Create(PENGGUNA b)
{
    if (ModelState.IsValid)
    {
        db.PENGGUNAS.Add(b); db.SaveChanges();
    }
    return RedirectToAction("DataPengguna");
}
```

Kode Sumber 4.4 ActionResult Create

```
[ActionName("DataPengguna")]
[OnAction(ButtonName = "Update")]
public ActionResult Edit(PENGGUNA b)
{
    if (ModelState.IsValid)
    {
        db.Entry(b).State = EntityState.Modified;
        db.SaveChanges();
    }
    return RedirectToAction("DataPengguna", new { id =
0 });
}
```

Kode Sumber 4.5 ActionResult Edit

```
public ActionResult DeletePengguna(int id)
{
}
```

```

        PENGGUNA p = db.PENGGUNAS.Find(id);
        db.PENGGUNAS.Remove(p);
        db.SaveChanges();
        return RedirectToAction("DataPengguna", new { id =
0 });
    }

```

Kode Sumber 4.6 ActionResult DeletePengguna

Untuk region Data Prestasi terdapat action method **ActionResult** yang mengembalikan dalam bentuk **View** dan action method **JsonResult** yang mengembalikan dalam bentuk Json. Dalam region Data Prestasi terdapat **ActionResult** TambahAnggota, **ActionResult** DeleteAnggota, **ActionResult** TambahDosen, **ActionResult** DeleteDosenPrestasi, **ActionResult** HistoriPrestasi, **ActionResult** DeleteHistoriPrestasi, **ActionResult** DataPrestasi, **ActionResult** Create, **ActionResult** Edit, dan **ActionResult** DeletePrestasi. Sedangkan untuk action method **JsonResult** terdapat dua yaitu **JsonResult** LoadListTahun2, **JsonResult** LoadListKegiatan, dan **JsonResult** LoadListCabang. Kode sumber untuk region Data Prestasi ditunjukkan pada Kode Sumber 9.4 Lampiran B.

Pada Laporan prestasi hampir sama dengan region lainnya, pada region ini terdapat *action method* **ActionResult** yang mengembalikan dalam bentuk **View** dan *action method* **JsonResult** yang mengembalikan dalam bentuk Json untuk menampilkan laporan dalam bentuk *chart*.

Pada region Rekomendasi Mahasiswa terdapat **ActionResult** RekomendasiMahasiswa yang berfungsi untuk menentukan daftar mahasiswa yang direkomendasikan untuk mengikuti satu cabang kegiatan, rekomendasi diukur dengan mencari nilai tertinggi dari mahasiswa yang dikelompokkan dalam rumpun mata kuliah. Sehingga dalam satu

rumpun mata kuliah terdapat nama-nama mahasiswa yang direkomendasikan. Kemudian dari setiap rumpun mata kuliah dipetakan dengan cabang kegiatan. Sebelum menentukan rekomendasi mahasiswa, harus dilakukan pemetaan antara rumpun mata kuliah dan cabang kegiatan. Proses pemetaan dilakukan pada region Pemetaan yang didalamnya terdapat `ActionResult Pemetaan`, `ActionResult TambahPemetaan`, `ActionResult TambahPemetaan`, dan `ActionResult DeletePemetaan`. Region pemetaan ditunjukkan pada Kode Sumber 9.5 Lampiran B. Setelah dilakukan pemetaan selanjutnya adalah proses untuk mencari rekomendasi mahasiswa yang ditunjukkan pada Kode Sumber 9.6 Lampiran B.

4.2.3.2. Kelas HomeController

Kelas controller ini digunakan untuk memuat kelas yang mengatur proses bisnis yang dapat diakses oleh umum. Pada HomeController ini terdapat `#region Entities` yang digunakan untuk mengakses database pada *entity*, `ActionResult Index` merupakan *action method* yang digunakan untuk mengembalikan *View* yang menampilkan berita, dan `ActionResult Post` untuk menampilkan berita pada *page list* tertentu.

4.2.3.3. Kelas MahasiswaController

Kelas controller ini digunakan untuk mengatur proses bisnis yang diakses oleh mahasiswa. `MahasiswaController` ini terbagi menjadi 6 *region* dan `ViewResult Index`. `ViewResult Index` untuk mengembalikan dalam bentuk *View*. 6 *region* yang terdapat dalam `MahasiswaController` adalah *region Entities*, *region Profil*, *region Prestasi*, *region Dana*, *region Nilai*, dan *region Berita*. Pada *region Entities* digunakan untuk mengakses database pada kelas model. Pada *region Profil*, *region Prestasi*, *region Dana*, *region Nilai*, dan *region Berita* mempunyai secara umum

mempunyai *action method* untuk menampilkan *View*, mengubah, dan menambah. Pada *region* Prestasi terdapat *action method JsonResult* untuk mengembalikan dalam bentuk Json yang digunakan untuk mengambil data tahun, *list* kegiatan, dan *list* cabang kegiatan.

4.2.3.4. Kelas UserController

Kelas controller ini digunakan untuk mengatur hak akses *user*. *UserController* digunakan untuk mengatur *Session user*. Jika user adalah admin maka akan diarahkan pada halaman *dashboard admin*, jika user mahasiswa maka akan diarahkan pada halaman *dashboard mahasiswa*.

BAB V

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tentang pengujian dan evaluasi aplikasi yang dikembangkan. Pengujian dilakukan untuk mengetahui fungsionalitas aplikasi. Hasil evaluasi menjelaskan mengenai rangkuman pengujian yang dilakukan pada aplikasi ini.

5.1. Lingkungan Pengujian

Lingkungan pengujian Tugas Akhir ini adalah sebagai berikut:

Prosesor	: Intel Core i3 2310M CPU @2.10 GHz
Memori	: 4.00 GB
Jenis Device	: Komputer Lenovo
Sistem Operasi	: Microsoft Windows 8.1 Pro 64-bit
Tools	: Visual Studio 2012 SQL Server 2012
Framework	: ASP.NET MVC 4 : Entity Framework 5
Server	: IIS 8 Express

5.2. Skenario Pengujian

Pada bab ini dijelaskan skenario pengujian aplikasi. Pengujian yang dilakukan adalah pengujian terhadap fungsionalitas aplikasi. Metode pengujian yang dilakukan adalah metode *black box*. Metode ini cenderung melihat pada hasil keluaran aplikasi.

5.2.1. Pengujian Rekomendasi *Simple Additive Weighting* Dengan *Microsoft Excel*

Untuk mengetahui kebenaran hasil perhitungan rekomendasi prestasi maka dilakukan perhitungan manual berdasarkan persamaan *Simple Additive Weighting* dengan

menggunakan *Microsoft Excel*. Berikut akan ditunjukkan matriks kriteria pada Gambar 5.1. Matriks kriteria adalah matriks yang diperoleh dari proses perhitungan nilai untuk setiap kriteria pada setiap alternatif yang ada. Matriks ini diperoleh dari *query* yang ditunjukkan pada Kode Sumber 5.1 dan hasil query ditunjukkan pada Gambar 5.1.

```
select a.TAHUN_KEGIATAN, a.NAMA_KEGIATAN,a.NAMA_CABANG,
SUM(a.PESERTA) as peserta, SUM(a.FINALIS) as finalis,
SUM(a.POIN_JUARA) as poin_juara, a.POIN_TINGKAT as
poin_tingkat from (
SELECT
KEGIATAN_PRESTASI.TAHUN_KEGIATAN,KEGIATAN_PRESTASI.NAMA_KEGIAT
AN,CABANG_KEGIATAN.NAMA_CABANG, TINGKAT_KEGIATAN.POIN_TINGKAT,

                PESERTA=CASE
STATUS_PRESTASI.NAMA_STATUS_PRESTASI
                WHEN 'PESERTA' THEN
SUM(PRESTASI.JUMLAH_ANGGOTA)
                ELSE 0
                END,

                FINALIS=CASE
STATUS_PRESTASI.NAMA_STATUS_PRESTASI
                WHEN 'FINALIS' THEN
SUM(PRESTASI.JUMLAH_ANGGOTA)
                ELSE 0
                END,

                POIN_JUARA=CASE
STATUS_PRESTASI.NAMA_STATUS_PRESTASI
                WHEN 'JUARA' THEN
SUM(PERINGKAT_JUARA.POIN_JUARA)
                ELSE 0
                END
from
PRESTASI
LEFT JOIN HISTORY_PRESTASI ON
(PRESTASI.ID_PRESTASI=HISTORY_PRESTASI.ID_PRESTASI)
LEFT JOIN STATUS_PRESTASI ON
(HISTORY_PRESTASI.ID_STATUS_PRESTASI=STATUS_PRESTASI.ID_STATUS
_PRESTASI)
LEFT JOIN CABANG_KEGIATAN ON
(PRESTASI.ID_CABANG=CABANG_KEGIATAN.ID_CABANG)
```

```

LEFT JOIN KEGIATAN_PRESTASI ON
(CABANG_KEGIATAN.ID_KEGIATAN_PRESTASI=KEGIATAN_PRESTASI.ID_KEGIATAN_PRESTASI)

LEFT JOIN PERINGKAT_JUARA ON
(PRESTASI.ID_PERINGKAT_JUARA=PERINGKAT_JUARA.ID_PERINGKAT_JUARA)

LEFT JOIN TINGKAT_KEGIATAN ON
(KEGIATAN_PRESTASI.ID_TINGKAT_KEGIATAN=TINGKAT_KEGIATAN.ID_TINGKAT_KEGIATAN)
group by KEGIATAN_PRESTASI.NAMA_KEGIATAN,
CABANG_KEGIATAN.NAMA_CABANG, KEGIATAN_PRESTASI.TAHUN_KEGIATAN,
STATUS_PRESTASI.NAMA_STATUS_PRESTASI,
TINGKAT_KEGIATAN.POIN_TINGKAT
) as a
group by a.NAMA_KEGIATAN, a.NAMA_CABANG, a.TAHUN_KEGIATAN,
a.POIN_TINGKAT

```

Kode Sumber 5.1 Query Matrix Kriteria

	TAHUN_KEGIATAN	NAMA_KEGIATAN	NAMA_CABANG	peserta	finalis	poin_juara	poin_tingkat
1	2014	GEMASTIK	Data Mining	3	3	1000	40
2	2014	GEMASTIK	Keamanan Jaringan	33	3	0	40
3	2014	GEMASTIK	Penrograman	6	0	0	40
4	2014	GEMASTIK	Pengembangan Game	12	0	0	40
5	2014	GEMASTIK	Perangkat Lunak	6	3	800	40
6	2014	GEMASTIK	Pernmainan Bisnis	3	3	0	40
7	2014	GEMASTIK	Piranti Cerdas	3	0	0	40
8	2014	INAICTA	Application : Perguruan Tinggi	3	0	0	40

Gambar 5.1 Matrix Kriteria Hasil Dari Query

Nilai kriteria peserta diperoleh dari perhitungan jumlah peserta yang mengikuti satu cabang kegiatan pada suatu kegiatan di tahun tertentu. Nilai Kriteria finalis diperoleh dari perhitungan jumlah finalis pada satu cabang kegiatan di suatu kegiatan pada tahun tertentu. Nilai kriteria poin_juara diperoleh dari perhitungan jumlah poin juara dari satu cabang kegiatan di suatu kegiatan pada tahun tertentu. Nilai kriteria poin_tingkat diperoleh dari poin tingkat yang sesuai dengan tingkat kegiatan.

Kemudian menentukan rating kecocokan setiap alternatif pada setiap kriteria, selanjutnya dibuat matriks keputusan berdasarkan kriteria. Dari setiap nilai kriteria dari hasil query pada Gambar 5.1 dicocokkan dengan bobot variabel. Hasil dari ranting kecocokan tersebut kemudian dibuat *query*

yang ditunjukkan pada Kode Sumber 4.3 dan hasil *query* dari matriks keputusan ini diperlihatkan pada gambar Gambar 5.2.

	TAHUN_KEGIATAN	NAMA_KEGIATAN	NAMA_CABANG	PESERTA	FINALIS	POIN_JUARA	POIN_TINGKAT
1	2014	GEMASTIK	Data Mining	0.20	0.20	1.00	0.80
2	2014	GEMASTIK	Keamanan Jaringan	0.80	0.20	0.00	0.80
3	2014	GEMASTIK	Pemrograman	0.20	0.00	0.00	0.80
4	2014	GEMASTIK	Pengembangan Game	0.40	0.00	0.00	0.80
5	2014	GEMASTIK	Perangkat Lunak	0.20	0.20	0.80	0.80
6	2014	GEMASTIK	Pernmainan Bisnis	0.20	0.20	0.00	0.80
7	2014	GEMASTIK	Piranti Cerdas	0.20	0.00	0.00	0.80
8	2014	INACTA	Application : Perguruan Tinggi	0.20	0.00	0.00	0.80

Gambar 5.2 Matirx Keputusan Hasil Dari Query

Dari matriks keputusan, kemudian dilakukan normalisasi matriks berdasarkan persamaan *Simple Additive Weighting* yang ditunjukkan pada persamaan 2.1. Pada persamaan tersebut diketahui untuk melakukan normalisasi harus menghitung nilai r_{ij} , dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i . Pada Tugas Akhir ini menggunakan r_{ij} dengan j adalah atribut keuntungan (*benefit*). Sehingga menggunakan persamaan $\frac{x_{ij}}{\max_i x_{ij}}$ jika j adalah atribut keuntungan (*benefit*). Dari Gambar 5.2 diketahui bahwa i adalah kolom dan j adalah baris. Hasil dari normalisasi ini kemudian ditunjukkan pada gambar Gambar 5.3. Kemudian dicari nilai V dari persamaan yang ditunjukkan pada 2.2. Untuk membandingkan dengan hasil rekomendasi prestasi dari sistem JUARA, maka dibuatlah perangkingan berdasarkan value seperti yang ditunjukkan pada Gambar 5.4. Berdasarkan hasil perhitungan manual dengan menggunakan excel diketahui bahwa hasil rekomendasi prestasi sama dengan hasil rekomendasi prestasi yang dihasilkan oleh aplikasi JUARA. Hasil rekomendasi prestasi hasil perhitungan oleh aplikasi JUARA diperlihatkan pada Gambar 5.5.

Normalisasi							
No.	TAHUN_KEGIATAN	NAMA_KEGIATAN	NAMA_CABANG	PESERTA	FINALIS	POIN_JUARA	POIN_TINGKAT
1	2014	GEMASTIK	Data Mining	0.25	1	1	1
2	2014	GEMASTIK	Keamanan Jaringan	1	1	0	1
3	2014	GEMASTIK	Pemrograman	0.25	0	0	1
4	2014	GEMASTIK	Pengembangan Game	0.5	0	0	1
5	2014	GEMASTIK	Perangkat Lunak	0.25	1	0.8	1
6	2014	GEMASTIK	Permainan Bisnis	0.25	1	0	1
7	2014	GEMASTIK	Piranti Cerdas	0.25	0	0	1
8	2014	INAICTA	Application : Perguruan Tinggi	0.25	0	0	1

Gambar 5.3 Normalisasi Matriks Keputusan

Normalisasi Terurut							Value
No.	TAHUN_KEGIATAN	NAMA_KEGIATAN	NAMA_CABANG	PESERTA	FINALIS	POIN_JUARA	
1	2014	GEMASTIK	Data Mining	0.25	1	1	2.45
2	2014	GEMASTIK	Perangkat Lunak	0.25	1	0.8	2.25
3	2014	GEMASTIK	Keamanan Jaringan	1	1	0	1.6
4	2014	GEMASTIK	Permainan Bisnis	0.25	1	0	1.45
5	2014	GEMASTIK	Pengembangan Game	0.5	0	0	1.07
6	2014	GEMASTIK	Pemrograman	0.25	0	0	1.065
7	2014	GEMASTIK	Piranti Cerdas	0.25	0	0	1.065
8	2014	INAICTA	Application : Perguruan Tinggi	0.25	0	0	1.065

Gambar 5.4 Rangkang Normalisasi Matriks Keputusan Dan Value

RANKING	TAHUN	KEGIATAN	CABANG	PESERTA	FINALIS	POIN JUARA	POIN TINGKAT	VALUE
1	2014	GEMASTIK	Data Mining	0.25	1.00	1.00	1.00	2.45
2	2014	GEMASTIK	Perangkat Lunak	0.25	1.00	0.80	1.00	2.25
3	2014	GEMASTIK	Keamanan Jaringan	1.00	1.00	0.00	1.00	1.60
4	2014	GEMASTIK	Permainan Bisnis	0.25	1.00	0.00	1.00	1.45
5	2014	GEMASTIK	Pengembangan Game	0.50	0.00	0.00	1.00	0.70
6	2014	GEMASTIK	Pemrograman	0.25	0.00	0.00	1.00	0.65
7	2014	GEMASTIK	Piranti Cerdas	0.25	0.00	0.00	1.00	0.65
8	2014	INAICTA	Application : Perguruan Tinggi	0.25	0.00	0.00	1.00	0.65

Gambar 5.5 Rekomendasi Prestasi Hasil Sistem Perhitungan Sistem JUARA

5.2.2. Pengujian Subjektif

Pegujian subjektif dilakukan untuk mengetahui penilaian dan tanggapan dari pengguna terhadap sejumlah aspek dari aplikasi sistem pendukung keputusan prestasi

terbaik. Pengujian subjektif dilakukan dengan menggunakan sistem Likert, suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Ada dua bentuk pertanyaan yang menggunakan Likert yaitu pertanyaan positif untuk mengukur minat positif, dan bentuk pertanyaan negatif untuk mengukur minat negatif. Pertanyaan positif diberi skor 5, 4, 3, 2, dan 1, sedangkan bentuk pertanyaan negatif diberi skor 1, 2, 3, 4, dan 5. Biasanya disediakan lima pilihan skala dengan format seperti:

1. Sangat Tidak Setuju (STS)
2. Tidak Setuju (TS)
3. Ragu-ragu (R)
4. Setuju (S)
5. Sangat setuju (SS) [8]

Pada kuisioner ini akan digunakan pengukuran minat positif terhadap aplikasi JUARA. Dalam pelaksanaan kuisioner ini melibatkan 2 dosen Teknik Informatika dan 6 tim dari mahasiswa yang mengikuti perlombaan Gemastik. Pengujian dilakukan dengan mempublikasikan website JUARA. Setelah pengguna membuka *link* untuk mencoba aplikasi JUARA, pengguna mengisi kuisioner dari penulis dalam bentuk *Google Doc form* dan kuisioner dalam bentuk *hardcopy*. Bentuk kuisioner dapat dilihat lebih jelas pada Lampiran C. Pada Tabel 5.1 dan Tabel 5.2 ditunjukkan data diri responden dari pihak mahasiswa dan dosen. Responden mahasiswa adalah mahasiswa yang telah mengikuti perlombaan. Sedangkan responden dosen adalah dosen yang berkepentingan terhadap pengolahan prestasi pada website JUARA.

Tabel 5.1 Data Diri Pengisi Kuisoner (Mahasiswa)

No	NRP	Nama	Keterangan
1	5111100076	Sindunuraga Rikarno Putra	Juara 1 Data Mining Gemastik 7
2	5110100047	Dmitri Yanno Mahayana	Juara II Pengembangan Perangkat Lunak Gemastik 7
3	5111100030	Shintya Dwiarisa Puspita	Finalis INAICTA
4	5110100020	Aida Muflichah	Peserta PIMNAS 2014
5	5111100142	Friska Ayu Listya Irawan	Penyisihan Pengembangan Game Gemastik 7
6	5111100080	Novandi Banitama	Finalis Permainan Bisnis Gemastik 7

Tabel 5.2 Data Diri Pengisi Kuisoner (Dosen)

No	NIP	Nama	Keterangan
1	19720528 199702 1 001	Dwi Sunaryono, S.Kom., M.Kom.	Sebagai admin master
2	19841016 200812 1 002	Radityo Anggoro, S.Kom.,M.Sc	Sebagai admin master

Pada Tabel 5.3 menunjukan hasil kuisoner dengan responden mahasiswa. Kemudian dari hasil kuisoner data diolah untuk mengetahui persentase dari masing-masing alternatif jawaban pada setiap soalnya, seperti yang ditunjukan pada Tabel 5.4.

Tabel 5.3 Hasil Kuisioner Penggunaan Website Juara oleh Mahasiswa

No	Jenis Penilaian	Penilaian				
		STS	TS	B	S	SS
1	Kemenarikan halaman info prestasi	0	1	1	4	0
2	Kemudahan proses memasukan info prestasi	0	2	1	3	0
3	Kemenarikan halaman permohonan dana	1	1	3	1	0
4	Kemudahan proses pengajuan permohonan dana	0	1	3	2	0
5	Kemenarikan halaman nilai kuliah	0	1	3	2	0
6	Kemudahan proses mengunggah nilai kuliah	0	0	1	4	1
7	Kemenarikan halaman mengunggah berita	0	1	3	1	1
8	Kemudahan proses mengunggah berita	0	0	1	5	0

Tabel 5.4 Presentase Alternatif Jawaban Pada Kuisioner Mahasiswa

Pertanyaan	Penilaian				
	STS	TS	B	S	SS
1	0	1	1	4	0
Presentase	0%	16,67%	16,67%	66,67%	0%
2	0	2	1	3	0
Presentase	0%	33,33%	16,67%	50%	0%
3	1	1	3	1	0
Presentase	0%	16,67%	50%	16,67%	0%
4	0	1	3	2	0
Presentase	0%	16,67%	50%	33,33%	0%
5	0	1	3	2	0
Presentase	0%	16,67%	50%	33,33%	0%
6	0	0	1	4	1

Pertanyaan	Penilaian				
	STS	TS	B	S	SS
Presentase 7	0%	0%	16,67%	66,67%	16,67%
	0	1	3	1	1
Presentase 8	0%	16,67%	50%	16,67%	16,67%
	0	0	1	5	0
Presentase	0%	0%	16,67%	83,33%	0%

Kemudian pada Tabel 5.5 ditunjukkan persentase interpretasi penilaian kuisioner untuk setiap pertanyaan kuisioner. Untuk mendapatkan interpretasi penilaian maka dilakukan langkah-langkah sebagai berikut:

1. Mencari skor ideal untuk pertanyaan yang diajukan kepada responden sebagai berikut:
 - a. Skor tertinggi: (SS) 5×6 responden = 30
 - b. Skor terendah: (STS) 1×6 responden = 6
2. Menghitung hasil interpretasi dengan: Total skor / Skor tertinggi $\times 100$
3. Melihat hasil interpretasi hasil penelitian dengan melihatnya dari persentase skala likert seperti yang ditunjukkan pada Tabel 5.6.

Tabel 5.5 Presentase Interpretasi Penilaian Kuisioner Mahasiswa

No	Jumlah Skor					Total Skor	Interpretasi
	STS (1)	TS (2)	B (3)	S (4)	SS (5)		
1	0	2	3	16	0	21	70%
2	0	4	3	12	0	19	63,33%
3	1	2	9	4	0	16	53,33%
4	0	2	9	8	0	19	63,33%
5	0	2	9	8	0	19	63,33%
6	0	0	3	16	5	24	80%
7	0	2	9	4	5	20	66,67%
8	0	0	3	20	0	23	76,67%

Tabel 5.6 Interpretasi Skor Hasil Penilaian Kuisioner Mahasiswa

No	STS	TS	B	S	SS	
	0%	20%	40%	60%	80%	100%
1						
2						
3						
4						
5						
6						
7						
8						
%				12.5%	87.5%	

Dari Tabel 5.6 dapat disimpulkan bahwa sebanyak 87.5 % responden mahasiswa setuju terhadap kemenarikan halaman info prestasi, permohonan dana, nilai kuliah, dan mengunggah berita. Selain kemenarikan, responden juga setuju terhadap kemudahan proses memasukan info prestasi, proses pengajuan permohonan dana, mengunggah nilai kuliah, dan mengunggah berita pada website JUARA.

Pada Tabel 5.7 menunjukan hasil kuisioner dengan responden mahasiswa. Kemudian dari hasil kuisioner data diolah untuk mengetahui persentase dari masing-masing alternatif jawaban pada setiap soalnya, seperti yang ditunjukan pada Tabel 5.8.

Tabel 5.7 Hasil Kuisioner Penggunaan Website Juara oleh Dosen

No	Jenis Penilaian	Penilaian				
		STS	TS	B	S	SS
1	Kemenarikan halaman manajemen data prestasi	0	0	1	1	0
2	Kemudahan melakukan manajemen data prestasi	0	0	1	1	0

No	Jenis Penilaian	Penilaian				
		STS	TS	B	S	SS
3	Terpenuhinya kebutuhan dokumentasi prestasi jurusan Teknik Informatika	0	0	0	2	0
4	Kemenarikan halaman manajemen permohonan dana	0	0	0	2	0
5	Kemudahan melakukan manajemen permohonan dana	0	0	0	2	0
6	Kesesuaian proses permohonan dana pada website JUARA dengan proses bisnis permohonan dana di jurusan Teknik Informatika	0	1	0	1	0
7	Kemenarikan halaman rekomendasi mahasiswa	0	0	1	1	0
8	Kemudahan melakukan manajemen rekomendasi mahasiswa	0	0	1	1	0
9	Kebermanfaatan hasil rekomendasi mahasiswa untuk mempersiapkan peserta lomba	0	1	0	0	1
10	Kemenarikan halaman rekomendasi prestasi	0	0	0	2	0
11	Kemudahan melakukan manajemen rekomendasi prestasi	0	0	0	2	0
12	Kebermanfaatan rekomendasi prestasi untuk membantu pemetaan fokus pembinaan perlombaan bagi mahasiswa Teknik Informatika	0	1	0	1	0
13	Kemenarikan halaman manajemen berita	0	0	0	1	1
14	Kemudahan melakukan manajemen berita	0	0	0	1	1

Tabel 5.8 Presentase Alternatif Jawaban Pada Kuisioner Dosen

Pertanyaan	Penilaian				
	STS	TS	B	S	SS
1	0	0	1	1	0
Presentase	0%	0%	50%	50%	0%
2	0	0	1	1	0
Presentase	0%	0%	50%	50%	0%
3	0	0	0	2	0
Presentase	0%	0%	0%	100%	0%
4	0	0	0	2	0
Presentase	0%	0%	0%	100%	0%
5	0	0	0	2	0
Presentase	0%	0%	0%	100%	0%
6	0	1	0	1	0
Presentase	0%	50%	0%	50%	0%
7	0	0	1	1	0
Presentase	0%	0%	50%	50%	0%
8	0	0	1	1	0
Presentase	0%	0%	50%	50%	0%
9	0	1	0	0	1
Presentase	0%	50%	0%	0%	50%
10	0	0	0	2	0
Presentase	0%	0%	0%	100%	0%
11	0	0	0	2	0
Presentase	0%	0%	0%	100%	0%
12	0	1	0	1	0
Presentase	0%	50%	0%	50%	0%
13	0	0	0	1	1
Presentase	0%	0%	0%	50%	50%
14	0	0	0	1	1
Presentase	0%	0%	0%	50%	50%

Kemudian pada Tabel 5.9 ditunjukkan persentase interpretasi penilaian kuisioner untuk setiap pertanyaan kuisioner. Untuk mendapatkan interpretasi penilaian maka dilakukan langkah-langkah sebagai berikut:

1. Mencari skor ideal untuk pertanyaan yang diajukan kepada responden sebagai berikut:
 - a. Skor tertinggi: (SS) 5×2 responden = 10
 - b. Skor terendah: (STS) 1×6 responden = 6
2. Menghitung hasil interpretasi dengan: Total skor / Skor tertinggi $\times 100$
3. Melihat hasil interpretasi hasil penelitian dengan melihatnya dari persentase skala likert seperti yang ditunjukkan pada Tabel 5.10.

Tabel 5.9 Presentase Interpretasi Penilaian Kuisioner Dosen

No	Jumlah Skor					Total Skor	Interpretasi
	STS (1)	TS (2)	B (3)	S (4)	SS (5)		
1	0	0	3	4	0	7	70%
2	0	0	3	4	0	7	70%
3	0	0	0	8	0	8	80%
4	0	0	0	8	0	8	80%
5	0	0	0	8	0	8	80%
6	0	2	0	4	0	6	80%
7	0	0	3	4	0	7	70%
8	0	0	3	4	0	7	70%
9	0	2	0	0	5	7	70%
10	0	0	0	8	0	8	80%
11	0	0	0	8	0	8	80%
12	0	2	0	4	0	6	60%
13	0	0	0	4	5	9	90%
14	0	0	0	4	5	9	90%

Tabel 5.10 Interpretasi Skor Hasil Penilaian Kuisioner Dosen

No	STS 0%	TS 20%	B 40%	S 60%	SS 80%	SS 100%
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
%			7.14%	78.57%	14.28%	

Dari Tabel 5.10 dapat disimpulkan bahwa respon terhadap website JUARA oleh responden dosen adalah 7.14 % netral, 78.57% setuju, dan 14.28% sangat setuju.

Pada Tabel 5.11 menunjukan hasil kuisioner dengan responden dosen dan mahasiswa. Kemudian dari hasil kuisioner data diolah untuk mengetahui persentase dari masing-masing alternatif jawaban pada setiap soalnya, seperti yang ditunjukan pada Tabel 5.12.

Tabel 5.11 Hasil Kuisioner Penggunaan Website Juara Secara Umum

No	Jenis Penilaian	Penilaian				
		STS	TS	B	S	SS
1	Kemenarikan website JUARA	0	3	3	2	0
2	Berita prestasi website JUARA informatif	0	0	2	5	1

Tabel 5.12 Presentase Alternatif Jawaban Pada Kuisioner umum

Pertanyaan	Penilaian				
	STS	TS	B	S	SS
1	0	3	3	2	0
Presentase	0%	37.5%	37.5%	25%	0%
2	0	0	2	5	1
Presentase	0%	0%	25%	62.5%	12.5%

Kemudian pada Tabel 5.13 ditunjukkan persentase interpretasi penilaian kuisioner untuk setiap pertanyaan kuisioner. Untuk mendapatkan interpretasi penilaian maka dilakukan langkah-langkah sebagai berikut:

1. Mencari skor ideal untuk pertanyaan yang diajukan kepada responden sebagai berikut:
 - a. Skor tertinggi: (SS) 5×8 responden = 40
 - b. Skor terendah: (STS) 1×8 responden = 8
2. Menghitung hasil interpretasi dengan: Total skor / Skor tertinggi $\times 100$
3. Melihat hasil interpretasi hasil penelitian dengan melihatnya dari persentase skala likert seperti yang ditunjukkan pada Tabel 5.14.

Tabel 5.13 Presentase Interpretasi Penilaian Kuisioner Umum

No	Jumlah Skor					Total Skor	Interpretasi
	STS (1)	TS (2)	B (3)	S (4)	SS (5)		
1	0	6	9	8	0	23	57.5%
2	0	0	6	20	5	31	77.5%

Tabel 5.14 Interpretasi Skor Hasil Penilaian Kuisioner Umum

No	STS		TS		B		S		SS	
	0%	20%	40%		60%		80%		100%	
1										
2										
%					50%		50%			

Dari Tabel 5.14 dapat disimpulkan bahwa respon terhadap website JUARA oleh responden dosen dan mahasiswa adalah 50% netral dan 50% setuju.

Pada Tabel 5.15 ditunjukkan hasil penilaian responden terhadap website JUARA dan Tabel 5.16 menunjukkan hasil testimoni dan pendapat responden terhadap website JUARA.

Tabel 5.15 Hasil Kuisoner Penilaian untuk JUARA

No	Nama	Kritik Saran
1	Sindunuraga Rikarno Putra	UX UI sangat kurang
2	Dmitri Yanno Mahayana	1. Sisi <i>frontend</i> atau <i>front office</i> sudah bagus, cukup ringan. yang jelas ini sudah keliatan kalau datanya tersimpan secara terstruktur. 2. Sisi <i>backend</i> atau <i>back office</i> , kurang. Harusnya memiliki tema yang sesuai dengan prestasi. Lalu kurang interaktif jadi kurang bagus, jadi kurang menarik.
3	Shintya Dwiarisa Puspita	Website ini cukup berpotensi untuk kemudahan dalam berkomunikasi dengan dosen atau lainnya mengenai permohonan dana untuk kegiatan lomba maupun forum tertentu. Untuk pengembangan lebih lanjut mungkin pada tahap tambah anggota ada filter pencariannya atau input data langsung saja, karena belum tentu satu kelompok adalah mahasiswa Teknik Informatika semua dan agar lebih mudah dalam mencari nama tertentu.
4	Aida Muflichah	Tampilan bisa lebih menarik untuk halaman awal, disertai daftar prestasi mungkin di <i>sidebar</i> kanan. Pada pengisian prestasi, lebih efisien jika dalam aktivitas pertama kali input bisa langsung memasukkan anggota, dosen, dan status kepesertaan. Kurang menambahkan informasi syarat finalis

No	Nama	Kritik Saran
		pada <i>form</i> pendanaan. Jika harus memasukkan satu per satu nilai kuliah, kesusahan untuk user, mungkin aplikasi bisa mengambil informasi matakuliah dengan nilai yang diambil dari <i>database</i> SIAKAD Integra.
5	Friska Ayu Listya Irawan	Form untuk menambahkan data lebih memudahkan jika diletakan di atas.
6	Novandi Banitama	Sebaiknya nama dosen tidak terbatas nama dosen dari Teknik Informatika ITS saja. Kolom nama dosen juga ditambahkan pilihan lain-lain, setelah itu diminta menginput nama dosennya di textbox yang disediakan.
7	Dwi Sunaryono, S.Kom., M.Kom.	Tampilan masih kaku
8	Radityo Anggoro, S.Kom.,M.Sc	Proposal dana harus ada fitur komentar untuk permohonan dana yang revisi. Pemetaan kompetensi mata kuliah dengan kriteria lomba harus ditingkatkan dan detail.

Tabel 5.16 Hasil Kuisioner Testimoni Atau Pendapat Tentang JUARA

No	Nama	Testimoni
1	Sindunuraga Rikarno Putra	Idenya bagus dan bermanfaat
2	Dmitri Yanno Mahayana	Bagus, jadi segala prestasi mahasiswa Teknik Informatika dapat disimpan dengan rapi di JUARA. Apalagi kalau sampai bisa diintegrasikan dengan sistem lain, contoh SIAKAD, jadi otomatis bisa <i>generate</i> skem.
3	Shintya Dwiarisa Puspita	Ide sederhana yang cukup bermanfaat

No	Nama	Testimoni
4	Aida Muflichah	Aplikasi Juara ini menjadi prionir manajemen prestasi di Teknik Informatika.
5	Friska Ayu Listya Irawan	Website dibuat lebih berwarna
6	Novandi Banitama	Kalau dari segi fungsionalitas, sudah benar.
7	Dwi Sunaryono, S.Kom., M.Kom.	-
8	Radityo Anggoro, S.Kom.,M.Sc	Baik dan bermanfaat bagi jurusan Teknik Informatika

5.2.3. Pengujian Fungsionalitas

Pada subbab ini dijelaskan mengenai skenario pengujian fungsional dari sistem JUARA. Pengujian difokuskan pada pengujian fungsional. Pengujian yang dilakukan dipresentasikan pada Tabel 5.17.

Tabel 5.17 Daftar Pengujian Fungsional

Kode Pengujian	Uji Coba	Status
Mengelola Data Prestasi		
P-0001	Menambah Data Prestasi	Berhasil
P-0002	Menghapus Data Prestasi	Berhasil
P-0003	Mengubah Data Prestasi	Berhasil
Rekomendasi Mahasiswa		
P-0004	Menambah Rumpun Mata Kuliah	Berhasil
P-0005	Menambah Mata Kuliah	Berhasil
P-0006	Menambah Jenis Nilai	Berhasil
P-0007	Menetapkan Pemetaan	Berhasil
P-0008	Melihat Rekomendasi Mahasiswa	Berhasil
Rekomendasi Prestasi		
P-0009	Menambah Kriteria	Berhasil

Kode Pengujian	Uji Coba	Status
P-0010	Menambah Variabel	Berhasil
P-0011	Memasukan Bobot Kriteria	Berhasil
P-0012	Melihat Rekomendasi Prestasi	Berhasil
Mengelola Permohonan Dana		
P-0013	Mengubah Permohonan Dana	Berhasil
Mengelola Berita		
P-0014	Menambah Berita	Berhasil
P-0015	Mengubah Berita	Berhasil
P-0016	Menghapus Berita	Berhasil
Mengunggah Info Prestasi		
P-0017	Menambah Info Prestasi	Berhasil
Mengajukan Permohonan Dana		
P-0018	Menambahkan Permohonan Dana	Berhasil
Mengunggah Nilai		
P-0019	Menambahkan Nilai	Berhasil
Mengunggah Berita		
P-0020	Menambahkan Berita	Berhasil

5.2.3.1. Menambah Data Prestasi (P-0001)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan data prestasi ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.18. Tampilan pengujian dapat dilihat pada Gambar 5.7 dan Gambar 5.6.

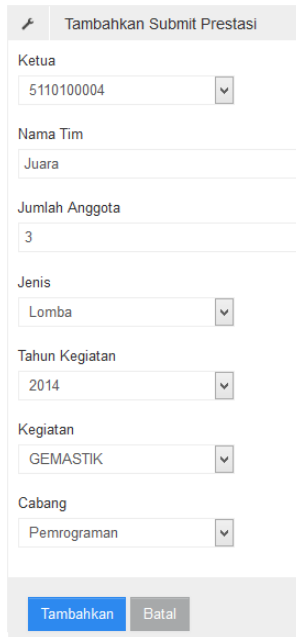
Tabel 5.18. Tabel Spesifikasi Pengujian Fungsional Menambah Data Prestasi ke Dalam Sistem

Kode Pengujian	P-0001
----------------	--------

Tujuan Pengujian		Memberi masukan data prestasi ke dalam sistem.		
Kondisi Awal		Administrator berada di halaman data prestasi.		
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Data prestasi	1. Memasukkan data prestasi ke textbo x 2. Tekan button tambahkan	Data prestasi dapat muncul pada halaman.	Data prestasi muncul dan siap untuk digunakan	Proses memberi masukan data prestasi ke dalam sistem berhasil
Kondisi Akhir	Administrator kembali halaman olah data prestasi dan data prestasi baru tersimpan.			

Tanggal	Anggota	Dosen	Nama Tim	Kegiatan	Cabang	Tahun	Status Prestasi	Juara	Actions
12/21/2014 12:11:13 PM	 1. IKHWAN FARISANDANI ★		Juara	GEMASTIK	Pemrograman	2014	 Peserta	Belum Juara	 

Gambar 5.6 Data Prestasi Tersimpan dalam Sistem



Tambahkan Submit Prestasi

Ketua
5110100004

Nama Tim
Juara

Jumlah Anggota
3

Jenis
Lomba

Tahun Kegiatan
2014

Kegiatan
GEMASTIK

Cabang
Pemrograman

Tambahkan Batal

Gambar 5.7 Memasukan Data Prestasi

5.2.3.2. Menghapus Data Prestasi (P-0002)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur menghapus data prestasi ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.19. Tampilan pengujian dapat dilihat pada Gambar 5.8, Gambar 5.9, dan Gambar 5.10.

Tabel 5.19. Tabel Spesifikasi Pengujian Fungsional Menghapus Data Prestasi ke Dalam Sistem

Kode Pengujian	P-0002
Tujuan Pengujian	Menghapus data prestasi dari dalam sistem.
Kondisi Awal	Administrator berada di halaman data prestasi.

Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Memilih tombol <i>icon</i> hapus pada data prestasi yang dipilih	1. Memilih tombol hapus pada data prestasi yang dipilih 2. Memilih tombol ok pada pesan konfirmasi	Data prestasi yang dipilih untuk dihapus akan dihapus dari sistem	Data prestasi yang dipilih untuk dihapus akan terhapus dari sistem	Proses menghapus data prestasi yang pilih berhasil
Kondisi Akhir	Administrator kembali halaman olah data prestasi dan data prestasi terhapus.			

9/25/2014 12:00:00 AM	1.1 KETUT MEGI TRISNAWAN ★		igem	GEMASTIK	Pengembangan Game	2014	Penyisihan	Belum Juara	
-----------------------	----------------------------	--	------	----------	-------------------	------	------------	-------------	--

Gambar 5.8 Memilih Data untuk Dihapus

Apakah anda yakin menghapus igem GEMASTIK?

OK

Cancel

Gambar 5.9 Pesan untuk Konfirmasi Hapus Data Prestasi

10

records per page

Search: igem

Tanggal	Anggota	Dosen	Nama Tim	Kegiatan	Cabang	Tahun	Status Prestasi	Juara	Actions
No matching records found									

Gambar 5.10 Data Prestasi Tim Igem Berhasil Dihapus Sistem

5.2.3.3. Mengubah Data Prestasi (P-0003)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur mengubah data prestasi ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.20. Tampilan pengujian dapat dilihat pada Gambar 5.11, Gambar 5.12, dan Gambar 5.13.

Tabel 5.20. Tabel Spesifikasi Pengujian Fungsional Mengubah Data Prestasi ke Dalam Sistem

Kode Pengujian	P-0003			
Tujuan Pengujian	Mengubah data prestasi dari dalam sistem.			
Kondisi Awal	Administrator berada di halaman data prestasi.			
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Memilih tombol <i>icon</i> edit pada data prestasi yang dipilih	<ol style="list-style-type: none"> Memilih tombol <i>icon</i> edit pada data prestasi yang dipilih Mengisi data baru prestasi Memilih tombol update 	Data prestasi yang dipilih akan diperbarui oleh sistem	Data prestasi yang dipilih akan diperbarui oleh sistem	Proses memperbarui data prestasi berhasil
Kondisi Akhir	Administrator kembali halaman olah data prestasi dan data prestasi diperbarui.			

Tanggal	Anggota	Dosen	Nama Tim	Kegiatan	Cabang	Tahun	Status Prestasi	Juara	Actions
12/21/2014 12:14:40 PM	 1. IKHWAN FARISANDANI ★ 2. USWATUN HASANA KUNIO	 1. Fajar Baskoro	Juara	GEMASTIK	Pemrograman	2014	 Pending	Belum Juara	 

Gambar 5.11 Memilih Data untuk Diperbarui

Edit Submit Prestasi

ID

1027

ID Pengguna

85

Nama Tim

Juara Editl

Jumlah Anggota

3

Jenis

Lomba

Tahun Kegiatan

2014

Kegiatan

GEMASTIK

Cabang

Pemrograman

Update

Batal

Gambar 5.12 Form untuk Mengubah Data Baru

12/21/2014 12:14:40 PM		1. IKHWAN FARISANDANI ★ 2. USWATUN HASANA KUNIO		1. Fajar Baskoro	Juara Editl	GEMASTIK	Pemrograman	2014		Penyisihan	Belum Juara	
---------------------------	--	--	--	------------------	-------------	----------	-------------	------	--	------------	-------------	--

Gambar 5.13 Hasil Mengubah Data Prestasi Berhasil

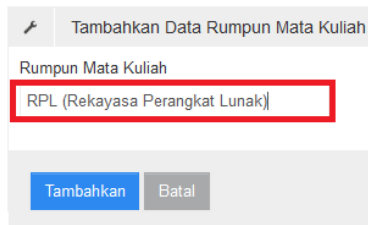
5.2.3.4. Menambah Rumpun Mata Kuliah (P-0004)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan rumpun mata kuliah ke dalam sistem. Penjelasan skenario

dapat dilihat di Tabel 5.21 . Tampilan pengujian dapat dilihat pada Gambar 5.14 dan Gambar 5.15.

Tabel 5.21. Tabel Spesifikasi Pengujian Fungsional Menambah Rumpun Mata Kuliah ke Dalam Sistem

Kode Pengujian	P-0004			
Tujuan Pengujian	Memberi masukan rumpun mata kuliah ke dalam sistem.			
Kondisi Awal	Administrator berada di halaman data rumpun .			
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Rumpun Mata Kuliah	1. Memasukkan rumpun mata kuliah ke textbox	Data rumpun mata kuliah dapat muncul	Data rumpun mata kuliah muncul dan siap untuk digunakan	Proses memberi masukan rumpun mata kuliah ke dalam sistem berhasil
Kondisi Akhir	2. Tekan button tambahkan	pada halaman.		
	Administrator kembali halaman data rumpun mata kuliah dan data rumpun mata kuliah yang baru tersimpan.			




Tambahkan Data Rumpun Mata Kuliah

Rumpun Mata Kuliah

RPL (Rekayasa Perangkat Lunak)

Tambahkan Batal

Gambar 5.14 Memasukan Rumpun Mata Kuliah

KBJ (Komputasi Berbasis Jaringan)	 
KCV (Komputasi Cerdas dan Visi)	 
Mi (Manajemen Informasi)	 
RPL (Rekayasa Perangkat Lunak)	 

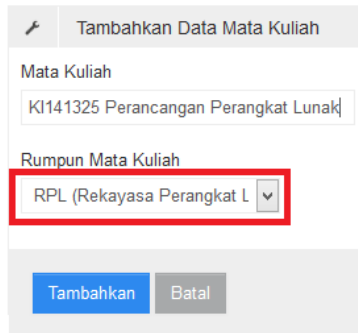
Gambar 5.15 Rumpun Mata Kuliah Tersimpan Dalam Sistem

5.2.3.5. Menambah Mata Kuliah (P-0005)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan mata kuliah ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.22. Tampilan pengujian dapat dilihat pada Gambar 5.16 dan Gambar 5.17.

Tabel 5.22. Tabel Spesifikasi Pengujian Fungsional Menambah Mata Kuliah ke Dalam Sistem

Kode Pengujian	P-0005			
Tujuan Pengujian	Memberi masukan mata kuliah ke dalam sistem.			
Kondisi Awal	Administrator berada di halaman data mata kuliah .			
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Mata Kuliah	1. Memasukkan mata kuliah ke textbox 2. Tekan button tambahkan	Data mata kuliah dapat muncul pada halaman.	Data mata kuliah muncul dan siap untuk digunakan	Proses memberi masukan mata kuliah ke dalam sistem berhasil
Kondisi Akhir	Administrator kembali halaman data mata kuliah dan data mata kuliah yang baru tersimpan.			



Gambar 5.16 Memasukan Data Mata Kuliah

Mata Kuliah	Rumpun Mata Kuliah	Actions
KI141325 Perancangan Perangkat Lunak	RPL (Rekayasa Perangkat Lunak)	 

Gambar 5.17 Data Mata Kuliah Baru Tersimpan Dalam Sistem


5.2.3.6. Menambah Jenis Kuliah(P-0006)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan jenis nilai ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.23. Tampilan pengujian dapat dilihat pada Gambar 5.18 dan Gambar 5.19.

Tabel 5.23. Tabel Spesifikasi Pengujian Fungsional Menambah Jenis Nilai ke Dalam Sistem

Kode Pengujian		P-0006		
Tujuan Pengujian		Memberi masukan jenis nilai ke dalam sistem.		
Kondisi Awal		Administrator berada di halaman data jenis nilai .		
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Jenis Nilai	1. Memasukkan jenis nilai ke textbox	Data jenis nilai dapat muncul pada halaman.	Data jenis nilai muncul dan siap untuk digunakan	Proses memberi masukan jenis nilai ke

	2. Tekan button tambahkan	dalam sistem berhasil
Kondisi Akhir	Administrator kembali halaman data jenis nilai dan data jenis nilai yang baru tersimpan.	



Tambahkan Data Jenis Nilai

Jenis Nilai

E

Poin

0

Tambahkan

Batal

Gambar 5.18 Memasukan Jenis Nilai

C	2.00	 
D	1.00	 
E	0.00	 

Gambar 5.19 Jenis Nilai Tersimpan dalam Sistem

5.2.3.7. Melakukan Pemetaan RMK (P-0007)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur melakukan pemetaan RMK ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.24. Tampilan pengujian dapat dilihat pada Gambar 5.20, Gambar 5.21, dan Gambar 5.22.


Tabel 5.24. Tabel Spesifikasi Pengujian Fungsional Menambah Jenis Nilai ke Dalam Sistem

Kode Pengujian	P-0007
Tujuan Pengujian	Memberi masukan rumpun mata kuliah pada setiap cabang kegiatan.

Kondisi Awal		Administrator berada di halaman pemetaan.		
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Rumpun mata kuliah untuk setiap cabang kegiatan	<ol style="list-style-type: none"> 1. Tekan <i>icon edit</i> pada kolom RMK pada cabang kegiatan yang dipilih 2. Memilih rumpun mata kuliah pada form tambah pemetaan 3. Tekan tombol tambahkan 	Pemetaan rumpun mata kuliah untuk setiap cabang kegiatan akan muncul pada halaman pemetaan.	Pemetaan rumpun mata kuliah untuk setiap cabang kegiatan muncul dan data siap digunakan	Proses melakukan pemetaan rumpun mata kuliah untuk setiap
Kondisi Akhir	Administrator kembali halaman pemetaan dan pemetaan tersimpan.			

Kegiatan	Tahun	Cabang	RMK
COMPTEST UI	2014	Intel® Real Sense™ Challenge	⊗
COMPTEST UI	2014	Sumo Robot Battle	⊗
COMPTEST UI	2014	Indie Game Ignite (GameDev)	⊗

Gambar 5.20 Memilih Cabang yang Akan Dipetakan


 Tambahkan Data Rumpun Mata Kuliah

Rumpun Mata Kuliah

RPL (Rekayasa Perangkat Lunak)

Tambahkan
 Batal

Gambar 5.21 Tambah RMK pada Cabang Kegiatan

Kegiatan	Cabang	RMK	Action
COMPTEST UI	Intel® Real Sense™ Challenge	RPL (Rekayasa Perangkat Lunak)	

Gambar 5.22 Berhasil Tambah Rumpun Mata Kuliah

5.2.3.8. Menampilkan Rekomendasi Mahasiswa (P-0008)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur menampilkan rekomendasi mahasiswa. Penjelasan skenario dapat dilihat di Tabel 5.25. Tampilan pengujian dapat dilihat pada Gambar 5.23.

Tabel 5.25. Tabel Spesifikasi Pengujian Fungsional Melihat Rekomendasi Mahasiswa ke Dalam Sistem

Kode Pengujian		P-0008		
Tujuan Pengujian		Melihat Rekomendasi Mahasiswa.		
Kondisi Awal		Administrator berada di halaman dashboard admin master.		
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
-	1. Memilih menu rekomendasi mahasiswa	Menampilkan rekomendasi mahasiswa	Rekomendasi mahasiswa dapat ditampilkan	Proses menampilkan rekomendasi mahasiswa berhasil
Kondisi Akhir	-			

KEGIATAN	CABANG	NRP	NAMA
GEMASTIK	Data Mining	5110100141	NURUL QAMAR
GEMASTIK	Data Mining	5111100080	NOVANDI BANITAMA
GEMASTIK	Pemrograman	5110100141	NURUL QAMAR
GEMASTIK	Pemrograman	5111100080	NOVANDI BANITAMA
GEMASTIK	Pengembangan Game	5110100141	NURUL QAMAR
GEMASTIK	Pengembangan Game	5110100150	LULUK EKO MAWIATI
GEMASTIK	Pengembangan Game	5111100142	FRISKA AYU LISTYA I
GEMASTIK	Pengembangan Game	5111100080	NOVANDI BANITAMA
GEMASTIK	Perangkat Lunak	5110100141	NURUL QAMAR
GEMASTIK	Perangkat Lunak	5110100150	LULUK EKO MAWIATI

Gambar 5.23 Tampilan Rekomendasi Mahasiswa

5.2.3.9. Menambah Kriteria (P-0009)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan kriteria ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.26. Tampilan pengujian dapat dilihat pada Gambar 5.24 dan Gambar 5.25.

Tabel 5.26. Tabel Spesifikasi Pengujian Fungsional Menambah Kriteria ke Dalam Sistem

Kode Pengujian		P-0009		
Tujuan Pengujian		Memberi masukan kriteria ke dalam sistem.		
Kondisi Awal		Administrator berada di halaman data kriteria.		
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Kriteria	1. Memasukkan kriteria ke textbox	Data kriteria dapat muncul pada halaman.	Data kriteria muncul dan siap untuk digunakan	Proses memberi masukan kriteria ke dalam sistem berhasil
	2. Tekan button tambahkan			
Kondisi Akhir	Administrator kembali halaman data kriteria dan data kriteria yang baru tersimpan.			

Tambahkan Data Kriteria

Kriteria

PENYELENGGARA_LOMBA

Tambahkan

Batal

Gambar 5.24 Memasukan Kriteria

Kriteria

JUMLAH_FINALIS

JUMLAH_PESERTA

PENYELENGGARA_LOMBA

POIN_KEJUARAAN

TINGKAT_LOMBA

Gambar 5.25 Kriteria Baru Tersimpan Dalam Sistem

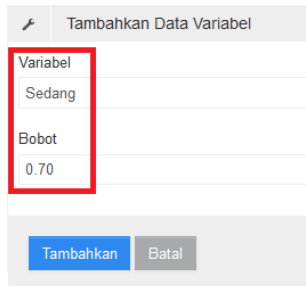
5.2.3.10. Menambah Variabel (P-0010)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan variabel ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.27. Tampilan pengujian dapat dilihat pada Gambar 5.26 dan Gambar 5.27.

Tabel 5.27. Tabel Spesifikasi Pengujian Fungsional Menambah Variabel ke Dalam Sistem

Kode Pengujian	P-0010
Tujuan Pengujian	Memberi masukan variabel ke dalam sistem.
Kondisi Awal	Administrator berada di halaman data variabel.

Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Variabel	1. Memasukkan variabel ke textbox 2. Tekan button tambahkan	Variabel dapat muncul pada halaman.	Variabel muncul dan siap untuk digunakan	Proses memberi masukan variabel ke dalam sistem berhasil
Kondisi Akhir	Administrator kembali halaman variabel dan variabel yang baru tersimpan.			



Gambar 5.26 Memasukan Variabel

Variabel	Bobot
Cukup	0.60
Rendah	0.40
Sangat Rendah	0.20
Sangat Tinggi	1.00
Sedang	0.70
Tinggi	0.80

Gambar 5.27 Variabel Baru Tersimpan Dalam Sistem

5.2.3.11. Menambah Bobot Kriteria (P-0011)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan

bobot kriteria ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.28. Tampilan pengujian dapat dilihat pada Gambar 5.28 dan Gambar 5.29.

Tabel 5.28. Tabel Spesifikasi Pengujian Fungsional Menambah Bobot Kriteria ke Dalam Sistem

Kode Pengujian	P-0011			
Tujuan Pengujian	Memberi masukan bobot kriteria ke dalam sistem.			
Kondisi Awal	Administrator berada di halaman bobot kriteria.			
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Bobot Kriteria	1. Pilih kriteria dengan variabel tertentu yang akan ditambahkan bobotnya	Bobot dapat ditampilkan pada halaman bobot kriteria.	Bobot muncul dan siap untuk digunakan	Proses memberi masukan bobot ke dalam sistem berhasil
	2. Tekan button edit pada kolom kriteria yang dipilih			
	3. Mengisi bobot pada kriteria dengan variabel tertentu tersebut			
	4. Tekan tombol <i>update</i>			

Kondisi Akhir	Administrator kembali halaman bobot kriteria dan bobot kriteria yang baru tersimpan.
---------------	--

Edit Data Bobot Kriteria

ID
21

Kriteria
JUMLAH_FINALIS

Variabel
Sangat Tinggi

Nilai Mulai
41

Nilai Batas
50

Update **Batal**

Gambar 5.28 Memasukan Bobot

Kriteria	Variabel	Nilai Mulai	Nilai Batas	Actions
JUMLAH_FINALIS	Sangat Tinggi	41	50	
JUMLAH_FINALIS	Tinggi	31	40	
JUMLAH_FINALIS	Cukup	21	30	

Gambar 5.29 Bobot Baru Tersimpan Dalam Sistem

5.2.3.12. Menampilkan Rekomendasi Prestasi (P-0012)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur menampilkan rekomendasi mahasiswa. Penjelasan skenario dapat dilihat di Tabel 5.29. Tampilan pengujian dapat dilihat pada Gambar 5.30.

Tabel 5.29. Tabel Spesifikasi Pengujian Fungsional Melihat Rekomendasi Mahasiswa ke Dalam Sistem

Kode Pengujian		P-0008		
Tujuan Pengujian		Melihat Rekomendasi Mahasiswa.		
Kondisi Awal		Administrator berada di halaman dashboard admin master.		
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
-	2. Memilih menu rekomendasi mahasiswa	Menampilkan rekomendasi mahasiswa	Rekomendasi mahasiswa dapat ditampilkan	Proses menampilkan rekomendasi mahasiswa berhasil
Kondisi Akhir	-			

KEGIATAN	CABANG	NRP	NAMA
GEMASTIK	Data Mining	5110100141	NURUL QAMAR
GEMASTIK	Data Mining	5111100080	NOVANDI BANITAMA
GEMASTIK	Pemrograman	5110100141	NURUL QAMAR
GEMASTIK	Pemrograman	5111100080	NOVANDI BANITAMA
GEMASTIK	Pengembangan Game	5110100141	NURUL QAMAR
GEMASTIK	Pengembangan Game	5110100150	LULUK EKO MAWATI
GEMASTIK	Pengembangan Game	5111100142	FRISKA AYU LISTYA I

Gambar 5.30 Tampilan Rekomendasi Mahasiswa

5.2.3.13. Mengubah Status Permohonan Dana (P-0013)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur mengubah status permohonan dana ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.30. Tampilan pengujian dapat dilihat pada Gambar 5.31, Gambar 5.32, Gambar 5.33, dan Gambar 5.34.

Tabel 5.30. Tabel Spesifikasi Pengujian Fungsional Mengubah Permohonan Dana ke Dalam Sistem

Kode Pengujian	P-0003			
Tujuan Pengujian	Mengubah status permohonan dana dari dalam sistem.			
Kondisi Awal	Administrator berada di halaman permohonan dana.			
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Memperbarui status permohonan dana	<ol style="list-style-type: none"> Memilih tombol <i>icon</i> edit pada permohonan dana yang dipilih Memilih status prestasi yang baru Memilih tombol update 	Status permohonan dana yang baru muncul	Status permohonan dana berhasil ditampilkan	Proses memperbarui status permohonan dana berhasil tersimpan dalam sistem
Kondisi Akhir	Administrator kembali halaman permohonan dana.			

Ketua	Nama Tim	Kegiatan	Cabang	Judul	File	Status Keuangan	Status Permohonan	Actions
MUHAMAD ARDHIHATA J	Dot AVI	GEMASTIK	Keamanaan Jaringan	Proposal Berangkat Final GEMASTIK	5112100094_13_201412072030.pdf	Finalis	Mengajukan	 

Gambar 5.31 Memilih Permohonan Dana yang Diubah

Edit Permohonan Dana

ID

3

File Proposal

No file selected5112100094_13_201412072030.pdf

Status

Mengajukan

Judul Proposal

Proposal Berangkat Final GEMASTIK

UpdateBatal

Gambar 5.32 Status Permohonan Dana Sebelum Diubah

Edit Permohonan Dana

ID

3

File Proposal

No file selected5112100094_13_201412072030.pdf

Status

Revisi

Judul Proposal

Proposal Berangkat Final GEMASTIK

UpdateBatal

Gambar 5.33 Status Permohonan Dana yang Akan Diperbarui

Ketua	Nama Tim	Kegiatan	Cabang	Judul	File	Status Kejuaraan	Status Permohonan	Actions
MUHAMMAD ARDHIWATA J	Dot AVI	GEMASTIK	Keamanan Jaringan	Proposal Berangkat Final GEMASTIK	5112100094_13_201412072030.pdf	Finalis	Revisi	

Gambar 5.34 Status Baru Permohonan Dana Tersimpan

5.2.3.14. Menambah Berita (P-0014)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan berita ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.31. Tampilan pengujian dapat dilihat pada Gambar 5.35 dan Gambar 5.36.

Tabel 5.31. Tabel Spesifikasi Pengujian Fungsional Menambah Berita ke Dalam Sistem

Kode Pengujian	P-0013			
Tujuan Pengujian	Memberi masukan berita ke dalam sistem.			
Kondisi Awal	Administrator berada di halaman manajemen berita.			
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Berita	1. Mengunggah foto berita, mengisi judul dan isi ke textbox, dan memilih status berita 2. Tekan button tambahkan	Berita dapat ditampilkan pada halaman berita.	Berita muncul pada halaman berita.	Proses menambah berita ke dalam sistem berhasil
Kondisi Akhir	Administrator kembali halaman manajemen berita dan berita yang baru tersimpan.			

Tambahkan Berita

Gambar
20140415_1.jpg

Judul
Informatika ITS Juara Imagine Cup 2014


Isi

Mahasiswa Teknik Informatika ITS menjuarai kompetisi Imagine Cup 2014 kategori World Citizenship. Tim yang beranggotakan Muhammad Rizky Habibi, Nurul Wakhidatul Ummah, Mentari Queen Glossyta, dan Fiandra Fatarany mempresentasikan aplikasi bernama "Cakra" yaitu aplikasi terapi autisme yang mereka kembangkan dan berhasil memukau penonton serta dewan juri yang memenuhi Epicentrum Walk, Jakarta. Dengan prestasi ini, mereka berhak untuk mewakili Indonesia di ajang World Semifinal.

Status
Draft

Tambahkan **Batal**

Gambar 5.35 Memasukan Berita

Informatika ITS Juara Imagine Cup 2014	Mahasiswa Teknik Informatika ITS menjuarai kompetisi Imagine Cup 2014 kategori World Citizenship. Tim yang beranggotakan Muhammad Rizky Habibi, Nurul Wakhidatul Ummah, Mentari Queen Glossyta, dan Fiandra Fatarany mempresentasikan aplikasi bernama "Cakra" yaitu aplikasi terapi autisme yang mereka kembangkan dan berhasil memukau penonton serta dewan juri yang memenuhi Epicentrum Walk, Jakarta. Dengan prestasi ini, mereka berhak untuk mewakili Indonesia di ajang World Semifinal.	12/21/2014 4:29:27 PM		Draft
--	--	-----------------------	---	--------------

Gambar 5.36 Berita Berhasil Tersimpan

5.2.3.15. Mengubah Berita (P-0015)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur mengubah berita ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.32. Tampilan pengujian dapat dilihat pada Gambar 5.37 dan Gambar 5.38.

Tabel 5.32. Tabel Spesifikasi Pengujian Fungsional Mengubah Berita ke Dalam Sistem

Kode Pengujian	P-0014			
Tujuan Pengujian	Mengubah berita dari dalam sistem.			
Kondisi Awal	Administrator berada di halaman manajemen berita.			
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Memilih tombol <i>icon</i> edit pada berita yang dipilih	<ol style="list-style-type: none"> Memilih tombol <i>icon</i> edit pada berita yang dipilih Mengisi data baru berita Memilih tombol update 	Berita yang dipilih akan diperbarui oleh sistem	Berita yang dipilih akan diperbarui oleh sistem	Proses memperbarui berita berhasil
Kondisi Akhir	Administrator kembali ke halaman manajemen berita dan berita diperbarui.			

Edit Berita

ID
5

Tanggal
12/21/2014 4:29:27 PM

Gambar
No file select... 201412211629_informatika-ITS-Juarai-Imagine-Cup-2014.jpg

Judul
Informatika ITS Juara Imagine Cup 2014




Isi

Mahasiswa Teknik Informatika ITS menjuarai kompetisi Imagine Cup 2014 kategori World Citizenship. Tim yang beranggotakan Muhammad Rizky Habibi, Nurul Wakhidatul Ummah, Mentari Queen Glossyta, dan Fiandra Fatarany mempresentasikan aplikasi bernama "Cakra" yaitu aplikasi terapi autisme yang mereka kembangkan dan berhasil memukau penonton serta dewan juri yang memenuhi Epicentrum Walk, Jakarta. Dengan prestasi ini, mereka berhak untuk mewakili Indonesia di ajang World Semifinal.

Status
Final

Update **Batal**

Gambar 5.37 Memasukan Data Baru Berita

Informatika ITS Juara Imagine Cup 2014	Mahasiswa Teknik Informatika ITS menjuarai kompetisi Imagine Cup 2014 kategori World Citizenship. Tim yang beranggotakan Muhammad Rizky Habibi, Nurul Wakhidatul Ummah, Mentari Queen Glossyta, dan Fiandra Fatarany mempresentasikan aplikasi bernama "Cakra" yaitu aplikasi terapi autisme yang mereka kembangkan dan berhasil memukau penonton serta dewan juri yang memenuhi Epicentrum Walk, Jakarta. Dengan prestasi ini, mereka berhak untuk mewakili Indonesia di ajang World Semifinal.	12/21/2014 4:29:27 PM		Final	 
--	--	-----------------------	--	--------------	---

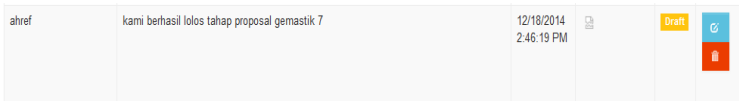
Gambar 5.38 Berita Baru Telah Tersimpan Dalam Sistem

5.2.3.16. Menghapus Berita (P-0016)

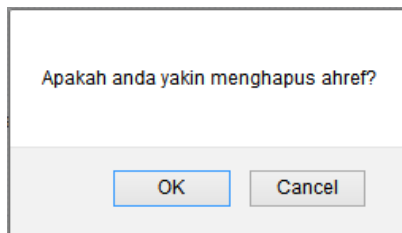
Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur menghapus berita ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.33. Tampilan pengujian dapat dilihat pada Gambar 5.39, Gambar 5.40, dan Gambar 5.41.

Tabel 5.33. Tabel Spesifikasi Pengujian Fungsional Menghapus Berita ke Dalam Sistem


Kode Pengujian Tujuan Pengujian Kondisi Awal		P-0015 Menghapus berita dari dalam sistem. Administrator berada di halaman data prestasi.		
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Memilih tombol <i>icon</i> hapus pada berita yang dipilih	1. Memilih tombol hapus pada berita yang dipilih 2. Memilih tombol ok pada pesan konfirmasi	Berita yang dipilih untuk dihapus akan dihapus dari sistem	Berita yang dipilih untuk dihapus akan terhapus dari sistem	Proses menghapus berita yang pilih berhasil
Kondisi Akhir	Administrator kembali ke halaman manajemen berita dan berita terhapus.			



Gambar 5.39 Memilih Data untuk Dihapus



Gambar 5.40 Pesan untuk Konfirmasi Hapus Data Prestasi

10  records per page

Search:

Judul	Berita	Tanggal	Gambar	Status	Actions
No matching records found					

Gambar 5.41 Data Prestasi Tim Igem Berhasil Dihapus Sistem

5.2.3.17. Menambah Info Prestasi (P-0017)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan info prestasi ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.34. Tampilan pengujian dapat dilihat pada Gambar 5.42 dan Gambar 5.43.

Tabel 5.34. Tabel Spesifikasi Pengujian Fungsional Menambah Info Prestasi ke Dalam Sistem

Kode Pengujian		P-0016		
Tujuan Pengujian		Memberi masukan info prestasi ke dalam sistem.		
Kondisi Awal		Mahasiswa berada di halaman info prestasi.		
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Data prestasi	1. Memasukkan info prestasi ke textbox	Info prestasi dapat muncul pada halaman.	Info prestasi muncul dan siap untuk digunakan	Proses memberi masukan info prestasi ke dalam sistem berhasil
	2. Tekan button tambahkan			
Kondisi Akhir	Mahasiswa kembali halaman olah data prestasi dan data prestasi baru tersimpan.			

Tambahkan Submit Prestasi

Nama Tim

Impian Budi

Jumlah Anggota

3

Jenis

Lomba

Tahun Kegiatan

2014

Kegiatan

GEMASTIK

Cabang

Pengembangan Game

Tambahkan

Batal

Gambar 5.42 Memasukan Info Prestasi

Ketua	Anggota	Dosen	Nama Tim	Kegiatan	Cabang	Status Prestasi	Juara	Actions
DMITRI YANNO MAHAYANA	1. DMITRI YANNO MAHAYANA 2. EKKY ARYA SUKARNO MAULANA 3. MUHAMMAD HANIF BUDIARTO	1. Dwi Sunaryono	LP	GEMASTIK	Perangkat Lunak	Juara	Juara 2	<div></div> <div></div>
DMITRI YANNO MAHAYANA	1. DMITRI YANNO MAHAYANA 2. MUHAMMAD HANIF BUDIARTO 3. MUHAMMAD REDHA		Impian Budi	GEMASTIK	Pengembangan Game	Pemilihan	Belum Juara	<div></div> <div></div>

Gambar 5.43 Info Prestasi Tersimpan dalam Sistem

5.2.3.18. Menambah Permohonan Dana (P-0018)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan permohonan dana ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.35. Tampilan pengujian dapat dilihat pada Gambar 5.44 dan Gambar 5.45.

Tabel 5.35. Tabel Spesifikasi Pengujian Fungsional Menambah Permohonan Dana ke Dalam Sistem

Kode Pengujian		P-0017		
Tujuan Pengujian		Memberi masukan permohonan dana ke dalam sistem.		
Kondisi Awal		Mahasiswa berada di halaman permohonan dana.		
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Permohonan Dana	1. Memilih tim pada dropdown	Permohonan dana dapat muncul pada halaman.	Permohonan dana muncul dan siap untuk digunakan.	Proses memberi masukan permohonan dana ke dalam sistem berhasil.
	2. Mengunggah berkas permohonan dana pada <i>fileinput</i>			
	3. Menulis judul berkas permohonan dana pada <i>textbox</i>			
	4. Tekan <i>button</i> tambahkan			
Kondisi Akhir	Mahasiswa kembali ke halaman permohonan dana dan permohonan dana baru tersimpan.			

Gambar 5.44 Memasukan Permohonan Dana

Ketua	Nama Tim	Kegiatan	Cabang	Judul	File	Status Prestasi	Status Permohonan	Actions
DMITRI YANNO MAHAYANA	LP	GEMASTIK	Perangkat Lunak	LP Gemastik	5110100047_1_201501071514.pdf	Juara	Mengajukan	

Gambar 5.45 Permohonan Dana Tersimpan dalam Sistem


5.2.3.19. Menambah Nilai Kuliah (P-0019)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan nilai kuliah ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.36. Tampilan pengujian dapat dilihat pada Gambar 5.46 dan Gambar 5.47.

Tabel 5.36. Tabel Spesifikasi Pengujian Fungsional Menambah Nilai ke Dalam Sistem

Kode Pengujian	P-0018			
Tujuan Pengujian	Memberi masukan nilai ke dalam sistem.			
Kondisi Awal	Mahasiswa berada di halaman menambah nilai.			
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan

Nilai	<div>1. Memilih rumpun mata kuliah pada dropdown RMK</div> <div>2. Memilih mata kuliah pada dropdown mata kuliah</div> <div>3. Memilih nilai kuliah pada dropdown</div> <div>4. Tekan button tambahkan</div>	Nilai dapat muncul pada halaman.	Nilai muncul dan siap untuk digunakan.	Proses memberi masukan nilai ke dalam sistem berhasil
Kondisi Akhir	Mahasiswa kembali ke halaman nilai dan nilai baru tersimpan.			



Tambahkan Nilai

RMK

RPL (Rekayasa Perangkat L

▼

Mata Kuliah

KI141325 Perancangan Pera

▼

Nilai

AB

▼

Tambahkan

Batal

Gambar 5.46 Memasukan Nilai Kuliah

RMK	Mata Kuliah	Nilai	Actions
RPL (Rekayasa Perangkat Lunak)	KI141325 Perancangan Perangkat Lunak	AB	 

Gambar 5.47 Nilai Kuliah Tersimpan dalam Sistem

5.2.3.20. Mengunggah Berita (P-0013)

Skenario pengujian ini untuk mengetahui fungsionalitas perangkat lunak untuk fitur memberi masukan berita ke dalam sistem. Penjelasan skenario dapat dilihat di Tabel 5.37. Tampilan pengujian dapat dilihat pada Gambar 5.48 dan Gambar 5.49.

Tabel 5.37. Tabel Spesifikasi Pengujian Fungsional Menambah Berita ke Dalam Sistem

Kode Pengujian	P-0019			
Tujuan Pengujian	Memberi masukan berita ke dalam sistem.			
Kondisi Awal	Mahasiswa berada di halaman manajemen berita.			
Data Masukan	Prosedur Pengujian	Hasil yang Diharapkan	Hasil yang Diperoleh	Kesimpulan
Berita	1. Mengunggah foto berita, mengisi judul dan isi ke textbox, dan memilih status berita	Berita dapat ditampilkan pada halaman berita.	Berita muncul pada halaman berita.	Proses menambah berita ke dalam sistem berhasil
Kondisi Akhir	2. Tekan button tambahkan	Mahasiswa kembali halaman manajemen berita dan berita yang baru tersimpan.		

Gambar 5.48 Memasukan Berita

Gambar 5.49 Berita Berhasil Tersimpan

Pengujian verifikasi ini bertujuan untuk melakukan

dengan kenyataan perhitungan manual dan data yang terdapat pada jurusan Teknik Informatika. Data yang digunakan adalah data dokumentasi prestasi pada kegiatan Gemastik 7, seperti yang ditunjukkan pada Gambar 5.50.

Perangkat Lunak				Peserta	Finalis	Poin Juara	Poin Tingkat
1	LP	5110100047	Omitri Yanno Mahayana	6	3	800	40
2	haminsatu	5111100091	Ahmad Hayam Brilian				
Pengembangan Game							
3	DG Tiga	5111100109	Muhammad Farhan Adha	12	0	0	40
4	The Diamonds	5110100133	Rizka Noviana Indriyani				
5	A href	5111100031	Helmy Satria Martha Putra				
6	Impian Budi	5110100202	Muhammad Hanif Budiarto				
Piranti Cerdas							
7	UTX	5111100147	R Aditya Brahmana	1	1	0	40
Keamanan Jaringan							
8	ARD	5112100020	Ripas Filqadar	33	3	0	40
9	BerburuDanMeramu	5112100095	Yusro Tsaqova				
10	BombJat	5111100012	Putu Wiramaswara Widya				
11	DEF	5111100145	Erick Hendra Putra Alwando				
12	dot AVI	5112100094	Muhamad Ardhinata J				
13	FRF	5112100001	Muhammad Fakhrian Noor				
14	IDC	5113100151	Danang Adi Nugroho				
15	JKT	5113100106	John Stephanus Peter				
16	KBP	5112100110	Karsten Ari Agathon				
17	Kipa	5113100150	Ibnu Prayogi				
18	Panitia	5113100135	Azkaa Khoiruddin				
Data Mining							
19	Heisenbug	5111100076	Sindunuraga Rikarno Putra	3	3	1000	40
Pemrograman							
20	CassavaKid	5111100095	Andy William	3	3	0	40
Permainan Bisnis							
21	rawon	5111100080	Novandi Banitama	3	3	0	40

Gambar 5.50 Data Prestasi Gemastik 7 Mahasiswa Teknik Informatika

Kemudian dari data prestasi pada Gambar 5.50 dibuatlah matriks kriteria. Matriks kriteria adalah matriks yang diperoleh dari proses perhitungan nilai untuk setiap kriteria pada setiap alternatif yang ada. Matriks ini diperoleh dari *query* yang ditunjukkan pada Kode Sumber 5.1 dan hasil *query* ditunjukkan pada Gambar 5.51.

	TAHUN_KEGIATAN	NAMA_KEGIATAN	NAMA_CABANG	peserta	finalis	poin_juara	poin_tingkat
1	2014	GEMASTIK	Data Mining	3	3	1000	40
2	2014	GEMASTIK	Keamanan Jaringan	33	3	0	40
3	2014	GEMASTIK	Pemrograman	3	3	0	40
4	2014	GEMASTIK	Pengembangan Game	12	0	0	40
5	2014	GEMASTIK	Perangkat Lunak	6	3	800	40
6	2014	GEMASTIK	Pernmainan Bisnis	3	3	0	40
7	2014	GEMASTIK	Piranti Cerdas	1	1	0	40

Gambar 5.51 Matrix Kriteria Hasil Dari Query

Kemudian menentukan rating kecocokan setiap alternatif pada setiap kriteria, selanjutnya dibuat matriks keputusan berdasarkan kriteria. Dari setiap nilai kriteria dari hasil query pada Gambar 5.1 dicocokkan dengan bobot variabel. Hasil dari ranting kecocokan tersebut kemudian dibuat *query* yang ditunjukkan pada Kode Sumber 4.3 dan hasil *query* dari matriks keputusan ini diperlihatkan pada gambar Gambar 5.52.

	TAHUN_KEGIATAN	NAMA_KEGIATAN	NAMA_CABANG	PESERTA	FINALIS	POIN_JUARA	POIN_TINGKAT
1	2014	GEMASTIK	Data Mining	0.20	0.20	1.00	0.80
2	2014	GEMASTIK	Keamanan Jaringan	0.80	0.20	0.00	0.80
3	2014	GEMASTIK	Pemrograman	0.20	0.20	0.00	0.80
4	2014	GEMASTIK	Pengembangan Game	0.40	0.00	0.00	0.80
5	2014	GEMASTIK	Perangkat Lunak	0.20	0.20	0.80	0.80
6	2014	GEMASTIK	Pernmainan Bisnis	0.20	0.20	0.00	0.80
7	2014	GEMASTIK	Piranti Cerdas	0.20	0.20	0.00	0.80

Gambar 5.52 Matirx Keputusan Hasil Dari Query

Dari matriks keputusan, kemudian dilakukan normalisasi matriks. Kemudian dicari nilai V dari persamaan yang ditunjukkan pada 2.2. Untuk membandingkan dengan hasil rekomendasi prestasi dari sistem JUARA, maka dibuatlah perbandingan berdasarkan value seperti yang ditunjukkan pada Gambar 5.53. Berdasarkan hasil perhitungan manual dengan menggunakan excel diketahui bahwa hasil rekomendasi prestasi sama dengan hasil rekomendasi prestasi yang dihasilkan oleh aplikasi JUARA. Hasil rekomendasi prestasi hasil perhitungan oleh aplikasi JUARA diperlihatkan pada Gambar 5.54.

Normalisasi Terurut							Value	
No.	TAHUN_KEGIATAN	NAMA_KEGIATAN	NAMA_CABANG	PESERTA	FINALIS	POIN_JUARA		POIN_TINGKAT
1	2014	GEMASTIK	Data Mining	0.25	1	1	1	2.45
5	2014	GEMASTIK	Pengembangan Game	0.25	1	0.8	1	2.25
2	2014	GEMASTIK	Perangkat Lunak	1	1	0	1	1.6
3	2014	GEMASTIK	Keamanan Jaringan	0.25	1	0	1	1.45
6	2014	GEMASTIK	Pemrograman	0.25	1	0	1	1.45
7	2014	GEMASTIK	Piranti Cerdas	0.25	1	0	1	1.45
4	2014	GEMASTIK	Permainan Bisnis	0.5	0	0	1	0.7

Gambar 5.53 Rangkang Normalisasi Matriks Keputusan Dan Value

RANKING	TAHUN	KEGIATAN	CABANG	PESERTA	FINALIS	POIN JUARA	POIN TINGKAT	VALUE
1	2014	GEMASTIK	Data Mining	0.25	1.00	1.00	1.00	2.45
2	2014	GEMASTIK	Perangkat Lunak	0.25	1.00	0.80	1.00	2.25
3	2014	GEMASTIK	Keamanan Jaringan	1.00	1.00	0.00	1.00	1.60
4	2014	GEMASTIK	Pemrograman	0.25	1.00	0.00	1.00	1.45
5	2014	GEMASTIK	Permainan Bisnis	0.25	1.00	0.00	1.00	1.45
6	2014	GEMASTIK	Piranti Cerdas	0.25	1.00	0.00	1.00	1.45
7	2014	GEMASTIK	Pengembangan Game	0.50	0.00	0.00	1.00	0.70

Gambar 5.54 Rekomendasi Prestasi Hasil Sistem Perhitungan Sistem JUARA

Dengan membawa hasil perbandingan antara hitungan manual dengan hitungan hasil sistem yang menggunakan data riil, dilakukan verifikasi kepada stakeholder yang berkepentingan terhadap sistem JUARA. Untuk studi kasus jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember ini, stakeholder yang berkepentingan adalah dosen penanggungjawab prestasi. Verifikasi dilakukan dengan mengajukan kuisioner dan menunjukkan hasil pengujian. Data responden untuk pengujian verifikasi pada Tabel 5.2. Sedangkan untuk pertanyaan kuisioner ditunjukkan pada Tabel 5.38.

Tabel 5.38 Pertanyaan Kuisioner untuk Pengujian Verifikasi

Pertanyaan
1. Apakah hasil rekomendasi sudah sesuai dengan kenyataan di lapangan? 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Biasa 4. Setuju 5. Sangat Setuju

Pada Tabel 5.39 menunjukan hasil kuisioner dengan responden mahasiswa. Kemudian dari hasil kuisioner data diolah untuk mengetahui persentase dari masing-masing alternatif jawaban pada setiap soalnya, seperti yang ditunjukan pada Tabel 5.40.

Tabel 5.39 Tabel Hasil Kuisioner untuk Pengujian Verifikasi

No	Jenis Penilaian	Penilaian				
		STS	TS	B	S	SS
1	Kesesuaian rekomendasi prestasi dengan kenyataan di lapangan	0	0	0	2	0

Tabel 5.40 Presentase Alternatif Jawaban Pada Kuisioner Pengujian Verifikasi

Pertanyaan	Penilaian				
	STS	TS	B	S	SS
1	0	0	0	2	0
Presentase	0%	0%	0%	100%	0%

Kemudian pada Tabel 5.41 ditunjukan persentase interpretasi penilaian kuisioner untuk setiap pertanyaan kuisioner. Untuk

mendapatkan interpretasi penilaian maka dilakukan langkah-langkah sebagai berikut:

1. Mencari skor ideal untuk pertanyaan yang diajukan kepada responden sebagai berikut:
 - a. Skor tertinggi: (SS) 5×2 responden = 10
 - b. Skor terendah: (STS) 1×2 responden = 2
2. Menghitung hasil interpretasi dengan: Total skor / Skor tertinggi $\times 100$
3. Melihat hasil interpretasi hasil penelitian dengan melihatnya dari persentase skala likert seperti yang ditunjukkan pada Tabel 5.42.

Tabel 5.41 Presentase Interpretasi Penilaian Kuisoner Verifikasi

No	Jumlah Skor					Total Skor	Interpretasi
	STS (1)	TS (2)	B (3)	S (4)	SS (5)		
1	0	0	0	8	0	8	80%

Tabel 5.42 Interpretasi Skor Hasil Penilaian Kuisoner Verifikasi

No	STS		TS		B		S		SS	
	0%	20%	40%		60%		80%		100%	
1										
%							100%			

Dari Tabel 5.42 dapat disimpulkan bahwa respon terhadap kesesuaian rekomendasi prestasi dengan data pada lapangan adalah 100% setuju.

BAB VI

KESIMPULAN DAN SARAN

Bab berikut menjelaskan terkait kesimpulan dan saran - saran yang didapatkan dari hasil proses pembuatan Tugas Akhir.

6.1. Kesimpulan

Dalam proses pengerjaan TA dari tahap analisis, perancangan, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Sistem pendokumentasian prestasi mahasiswa jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya berhasil dibuat dengan fitur-fitur seperti dokumentasi prestasi, rekomendasi prestasi, rekomendasi mahasiswa, dan permohonan dana.
2. Metode *Fuzzy Multi-Attribute Decision Making Simple Additive Weighting* berhasil diterapkan untuk menentukan alternatif-alternatif prestasi terbaik jurusan Teknik Informatika atau rekomendasi prestasi yang diharapkan akan membantu dalam pemetaan fokus pembinaan prestasi.

6.2. Saran

1. Pemetaan kompetensi mata kuliah dengan kriteria lomba harus ditingkatkan untuk mendapatkan rekomendasi mahasiswa yang akan diikutsertakan dalam kegiatan prestasi.
2. Integrasi antara SIAKAD dengan JUARA untuk mempermudah pemenuhan kebutuhan data yang berhubungan dengan prestasi.
3. Pada sistem ini bisa ditambahkan fitur komentar untuk permohonan dana yang berstatus revisi, fitur histori pengajuan permohonan dana, dan fitur *template* proposal permohonan dana.

LAMPIRAN A. ANALISIS DAN PERANCANGAN SISTEM

8.1. Skenario Kasus Penggunaan

Tabel 8.1 Skenario Mengelola Data Prestasi

Nama Kasus Penggunaan	Mengelola Dokumentasi Prestasi
Nomor	UC02
Deskripsi	Admin menambahkan data prestasi Selain itu admin juga dapat mengubah dan menghapus data prestasi.
Aktor	Admin
Kondisi Awal	Admin telah masuk dalam halaman <i>dashboard</i> admin.
Alur Normal	1. Admin memilih pilihan data prestasi
	2. Sistem menampilkan daftar data prestasi
	3. Admin memilih pilihan edit data prestasi A1. Admin memilih pilihan tambah data prestasi A2. Admin memilih pilihan hapus data prestasi
	4. Admin memperbarui data prestasi
	5. Admin menyetujui <i>edit</i> data prestasi selesai
Alur Alternatif	A1. Admin memilih pilihan tambah data prestasi A1.1 Admin memilih NRP ketua B1.1 Admin menambah NRP B1.2 Menuju alur A1.1 A1.2 Admin menambah nama tim A1.3 Admin menambah jumlah anggota A1.4 Admin memilih anggota prestasi B1.1 Admin menambah anggota B1.2 Menuju alur nomor A1.4

	<p>A1.5 Admin memilih dosen B1.1 Admin menambah dosen B1.2 Menuju alur nomor A1.5</p> <p>A1.6 Admin memilih jenis kegiatan B1.1 Admin menambah jenis kegiatan B1.2 Menuju alur nomor A1.6</p> <p>A1.7 Admin memilih tahun B1.1 Admin menambah kegiatan B1.2 Menuju alur nomor A1.7</p> <p>A1.8 Admin memilih kegiatan B1.1 Admin menambah kegiatan B1.2 Menuju alur nomor A1.8</p> <p>A1.9 Admin memilih cabang kegiatan B1.1 Admin menambah cabang kegiatan B1.2 Menuju alur nomor A1.9</p> <p>A1.10 Admin memilih nama status kejuaraan B1.1 Admin menambah status kejuaraan B1.2 Menuju alur nomor A1.10</p> <p>A1.11 Admin memilih tingkat juara B1.1 Admin menambah tingkat kejuaraan B1.2 Menuju alur nomor A1.11</p> <p>A2. Admin memilih pilihan hapus data prestasi A2.1 Admin memilih hapus data prestasi A2.2 Admin menyetujui hapus data prestasi</p>
Kondisi Akhir	Data prestasi berhasil ditambahkan dalam <i>database</i> atau berhasil diubah.
Lainnya	Aktor pada skenario kasus penggunaan ini adalah Admin yang mempunyai role Admin Master.

Tabel 8.2 Skenario Mengelola Data Cabang Kegiatan

Nama Kasus Penggunaan	Mengelola Data Cabang Kegiatan
Nomor	UC03
Deskripsi	Admin menambahkan data cabang kegiatan. Selain itu admin juga dapat mengubah dan menghapus data cabang kegiatan.
Aktor	Admin
Kondisi Awal	Admin telah masuk dalam halaman <i>dashboard</i> admin.
Alur Normal	1. Admin memilih pilihan data cabang kegiatan
	2. Sistem menampilkan daftar data cabang kegiatan
	3. Admin memilih pilihan edit data cabang kegiatan A1. Admin memilih pilihan tambah data cabang kegiatan A2. Admin memilih pilihan hapus data cabang kegiatan
	4. Admin memperbarui data cabang kegiatan
	5. Admin menyetujui <i>edit</i> data cabang kegiatan
Alur Alternatif	A1. Admin memilih pilihan tambah cabang kegiatan A1.1 Admin memilih jenis kegiatan B1.1 Admin menambah jenis kegiatan B1.2 Menuju alur A1.1 A1.2 Admin memilih tahun kegiatan B1.1 Admin menambah kegiatan B1.2 Menuju alur nomor A1.7 A1.3 Admin memilih kegiatan B1.1 Admin menambah kegiatan C1.1 Admin menambah nama kegiatan

	C1.2 Admin menambah tahun kegiatan C1.3 Admin memilih tingkat kegiatan D1.1 Admin menambah tingkat kegiatan D1.2 Menuju Alur C1.3 C1.4 Menuju alur A1.1 B1.4 Menuju alur A1.3 A1.3 Admin memasukan nama cabang kegiatan A1.4 Admin menyetujui penambahan cabang kegiatan A2. Admin memilih pilihan hapus cabang kegiatan A2.1 Admin memilih hapus cabang kegiatan A2.2 Admin menyetujui hapus cabang kegiatan
Kondisi Akhir	Data kegiatan berhasil ditambahkan dalam <i>database</i> atau berhasil diubah.
Lainnya	Aktor pada skenario kasus penggunaan ini adalah Admin yang mempunyai role Admin Master.

Tabel 8.3 Skenario Mengelola Permohonan Dana

Nama Kasus Penggunaan	Mengelola Permohonan Dana
Nomor	UC04
Deskripsi	Admin menambahkan permohonan dana. Selain itu admin juga dapat mengubah dan menghapus permohonan dana.
Aktor	Admin
Kondisi Awal	Admin telah masuk dalam halaman <i>dashboard</i> admin.
Alur Normal	1. Admin memilih pilihan permohonan dana

	2. Sistem menampilkan daftar permohonan dana
	3. Admin memilih pilihan edit permohonan dana A1. Admin memilih pilihan tambah permohonan dana A2. Admin memilih pilihan hapus permohonan dana
	4. Admin memperbarui permohonan dana
	5. Admin menyetujui <i>edit</i> permohonan dana
Alur Alternatif	A1. Admin memilih pilihan tambah permohonan dana A1.1 Admin memilih nama tim B1.1 Admin menambah data prestasi B1.2 Menuju alur A1.1 A1.2 Admin menambah <i>file</i> proposal permohonan dana A1.3 Admin memilih status permohonan dana B1.1 Admin menambah status permohonan dana B1.2 Menuju alur A1.3 A1.4 Admin menambah judul proposal A2. Admin memilih pilihan hapus permohonan dana A2.1 Admin memilih hapus permohonan dana A2.2 Admin menyetujui hapus permohonan dana
Kondisi Akhir	Data permohonan dana berhasil ditambahkan dalam <i>database</i> atau berhasil diubah.
Lainnya	Aktor pada skenario kasus penggunaan ini adalah Admin yang mempunyai role Admin Master dan Admin Permohonan Dana

Tabel 8.4 Skenario Melihat Rekomendasi Mahasiswa

Nama Kasus Penggunaan	Melihat Rekomendasi Mahasiswa
Nomor	UC06
Deskripsi	Admin dapat melihat daftar nama mahasiswa yang direkomendasikan untuk mengikuti satu kegiatan berdasarkan cabang kegiatan. Nama mahasiswa ini diperoleh dari perjumlahan nilai mahasiswa berdasarkan rumpun mata kuliah.
Aktor	Admin
Kondisi Awal	Admin telah masuk dalam halaman <i>dashboard</i> admin.
Alur Normal	<ol style="list-style-type: none"> 1. Admin memilih rekomendasi mahasiswa 2. Sistem menampilkan rekomendasi mahasiswa <ol style="list-style-type: none"> A1. Sistem tidak menampilkan rekomendasi mahasiswa
Alur Alternatif	<ol style="list-style-type: none"> A1. Sistem tidak menampilkan rekomendasi mahasiswa <ol style="list-style-type: none"> A1.1 Memilih rumpun mata kuliah <ol style="list-style-type: none"> B1.1 Menambah rumpun mata kuliah B1.2 Menuju alur A1.1 A1.2 Memilih mata kuliah <ol style="list-style-type: none"> B1.1 Menambah mata kuliah B1.2 Menuju alur A1.2 A1.3 Memilih jenis nilai <ol style="list-style-type: none"> B1.1 Menambah jenis nilai B1.2 Menuju alur A1.3 A1.4 Pemetaan cabang kegiatan dengan rumpun mata kuliah <ol style="list-style-type: none"> B1.1 Menuju alur A1.1 B1.2 Menuju Alur A1.4
Kondisi Akhir	Sistem menampilkan rekomendasi mahasiswa

Lainnya	Aktor pada skenario kasus penggunaan ini adalah Admin yang mempunyai role Admin Master.
----------------	---

Tabel 8.5 Skenario Melihat Rekomendasi Prestasi

Nama Kasus Penggunaan	Melihat Rekomendasi Mahasiswa
Nomor	UC07
Deskripsi	Admin dapat melihat daftar cabang kegiatan berdasarkan ranjing prestasi terbaik.
Aktor	Admin
Kondisi Awal	Admin telah masuk dalam halaman <i>dashboard</i> admin.
Alur Normal	1. Admin memilih pilihan rekomendasi prestasi
	2. Sistem menampilkan rekomendasi prestasi A1. Sistem tidak menampilkan rekomendasi prestasi
Alur Alternatif	A1. Sistem tidak menampilkan rekomendasi prestasi A1.1 Mengubah nilai <i>range</i> untuk setiap bobot variabel kriteria B1.1 Menambahkan kriteria dan variabel B1.2 Menuju alur A1.1
Kondisi Akhir	Sistem menampilkan rekomendasi prestasi
Lainnya	Aktor pada skenario kasus penggunaan ini adalah Admin yang mempunyai role Admin Master.

Tabel 8.6 Skenario Mengelola Berita

Nama Kasus Penggunaan	Mengelola Berita
Nomor	UC08
Deskripsi	Admin menambahkan berita. Selain itu admin juga dapat mengubah dan menghapus data kegiatan.

Aktor	Admin
Kondisi Awal	Admin telah masuk dalam halaman <i>dashboard</i> admin.
Alur Normal	1. Admin memilih pilihan manajemen berita
	2. Sistem menampilkan daftar berita
	3. Admin memilih pilihan edit berita A1. Admin memilih pilihan tambah berita A2. Admin memilih pilihan hapus berita
	4. Admin memperbarui berita
	5. Admin menyetujui <i>edit</i> berita
Alur Alternatif	A1. Admin memilih pilihan tambah berita A1.1 Admin memilih gambar A1.2 Admin memasukan judul berita A1.3 Admin memasukan isi berita A1.4 Admin memilih status berita B1.1 Admin menambah status berita B1.2 Menuju alur A1.4 A1.5 Menyetujui menambah berita A2. Admin memilih pilihan hapus berita A2.1 Admin memilih hapus berita A2.2 Admin menyetujui hapus berita
Kondisi Akhir	Berita berhasil ditambahkan dalam <i>database</i> atau berhasil diubah.
Lainnya	Aktor pada skenario kasus penggunaan ini adalah Admin yang mempunyai role Admin Master.

Tabel 8.7 Skenario Mengunggah Info Prestasi

Nama Kasus Penggunaan	Mengunggah Info Prestasi
Nomor	UC09
Deskripsi	Mahasiswa menambahkan info prestasi. Selain itu mahasiswa juga dapat mengubah dan menghapus info prestasi.

Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah masuk dalam halaman <i>dashboard</i> mahasiswa.
Alur Normal	1. Mahasiswa memilih <i>submit</i> info prestasi
	2. Mahasiswa memilih pilihan edit info prestasi A1. Mahasiswa memilih pilihan tambah info prestasi A2. Mahasiswa memilih pilihan hapus info prestasi
	3. Mahasiswa memperbarui info prestasi
	4. Mahasiswa menyetujui <i>edit</i> info prestasi
Alur Alternatif	A1. Mahasiswa memilih pilihan tambah info prestasi A1.1 Mahasiswa memasukan nama tim A1.2 Mahasiswa memasukan jumlah anggota tim A1.3 Mahasiswa memasukan data anggota A1.4 Mahasiswa memasukan dosen pembimbing A1.5 Mahasiswa memilih jenis kegiatan A1.6 Mahasiswa memilih kegiatan A1.7 Mahasiswa memilih cabang kegiatan A1.8 Mahasiswa memilih status prestasi A1.9 Mahasiswa memilih tingkat kejuaraan A1.10 Menyetujui menambah info prestasi A2. Mahasiswa memilih pilihan hapus info prestasi A2.1 Mahasiswa memilih hapus info prestasi

	A2.2 Mahasiswa menyetujui hapus info prestasi
Kondisi Akhir	Data info prestasi berhasil ditambahkan dalam <i>database</i> atau berhasil diubah.

Tabel 8.8 Skenario Mengunggah Nilai

Nama Kasus Penggunaan	Mengunggah Nilai
Nomor	UC010
Deskripsi	Mahasiswa menambahkan nilai. Selain itu mahasiswa juga dapat mengubah dan menghapus nilai.
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah masuk dalam halaman <i>dashboard</i> mahasiswa.
Alur Normal	1. Mahasiswa memilih pilihan mengunggah nilai
	2. Sistem menampilkan daftar nilai
	3. Mahasiswa memilih pilihan edit nilai A1. Mahasiswa memilih pilihan tambah nilai A2. Mahasiswa memilih pilihan hapus nilai
	4. Mahasiswa memperbarui nilai
	5. Mahasiswa menyetujui <i>edit</i> nilai
Alur Alternatif	A1. Mahasiswa memilih pilihan tambah nilai A1.1 Mahasiswa memasukan rumpun mata kuliah A1.2 Mahasiswa memasukan mata kuliah A1.3 Mahasiswa memasukan nilai A2. Mahasiswa memilih pilihan hapus nilai A2.1 Mahasiswa memilih hapus nilai A2.2 Mahasiswa menyetujui hapus nilai
Kondisi Akhir	Data nilai berhasil ditambahkan dalam <i>database</i> atau berhasil diubah.

Tabel 8.9 Skenario Mengunggah Permohonan Dana

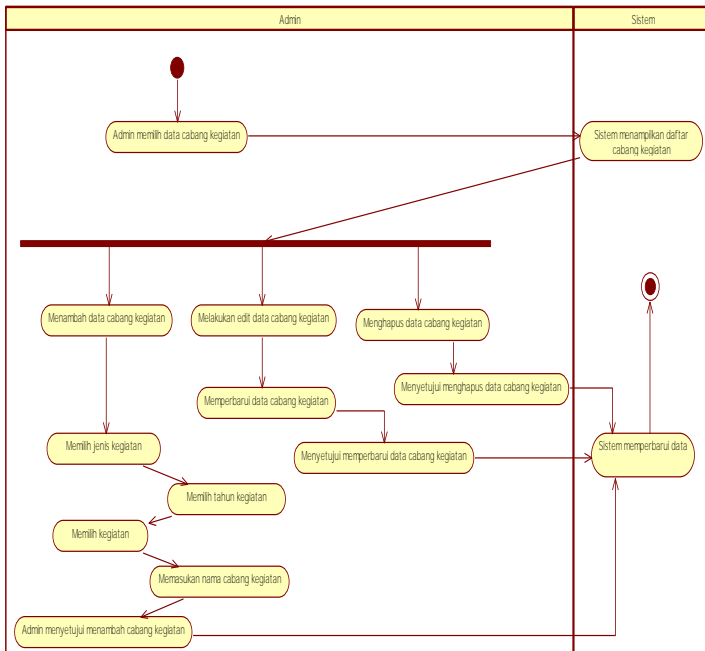
Nama Kasus Penggunaan	Mengunggah Permohonan Dana
Nomor	UC11
Deskripsi	Mahasiswa menambahkan permohonan dana. Selain itu mahasiswa juga dapat mengubah dan menghapus data permohonan dana.
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah masuk dalam halaman <i>dashboard</i> mahasiswa.
Alur Normal	1. Mahasiswa memilih pilihan mengunggah permohonan dana
	2. Sistem menampilkan daftar permohonan dana A1. Sistem tidak menampilkan daftar permohonan dana
	3. Mahasiswa memilih pilihan edit permohonan dana A2. Mahasiswa memilih pilihan tambah permohonan dana A3. Mahasiswa memilih pilihan hapus permohonan dana
	4. Mahasiswa memperbarui permohonan dana
	5. Mahasiswa menyetujui <i>edit</i> permohonan dana
Alur Alternatif	A1. Sistem tidak menampilkan daftar permohonan dana A2.1 Mahasiswa memilih pilihan menambah permohonan dana A2.1 Mahasiswa memilih nama tim B1.1 Mengubah status prestasi menjadi finalis B1.2 Menuju alur A2.1 A2.3 Mengunggah <i>file</i> proposal permohonan dana

	A2.4 Mahasiswa memasukan judul proposal permohonan dana A2.5 Mahasiswa menyetujui penambahan permohonan dana A3. Mahasiswa memilih pilihan menghapus permohonan dana A3.1 Mahasiswa memilih pilihan hapus permohonan dana A3.2 Mahasiswa menyetujui untuk menghapus permohonan dana
Kondisi Akhir	Data permohonan dana berhasil ditambahkan dalam <i>database</i> atau berhasil diubah.

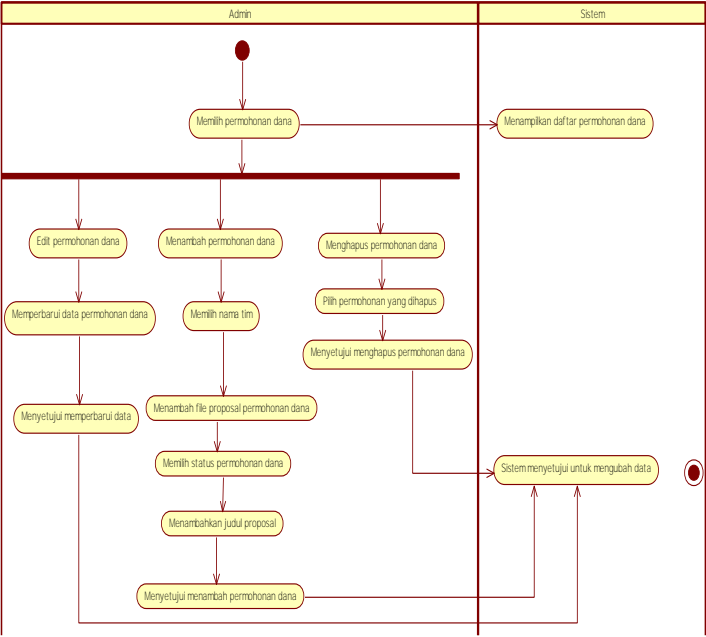
Tabel 8.10 Skenario Mengelola Berita

Nama Kasus Penggunaan	Mengunggah Berita
Nomor	UC12
Deskripsi	Mahasiswa menambahkan berita. Selain itu mahasiswa juga dapat mengubah dan menghapus berita.
Aktor	Mahasiswa
Kondisi Awal	Mahasiswa telah masuk dalam halaman <i>dashboard</i> mahasiswa.
Alur Normal	1. Mahasiswa memilih pilihan mengunggah berita 2. Sistem menampilkan daftar berita 3. Mahasiswa memilih pilihan edit berita A1. Mahasiswa memilih pilihan tambah berita A2. Mahasiswa memilih pilihan hapus berita
	4. Mahasiswa memperbarui berita
	5. Mahasiswa menyetujui <i>edit</i> berita
Alur Alternatif	A1. Mahasiswa memilih pilihan tambah berita A1.1 Mahasiswa memilih gambar A1.2 Mahasiswa memasukan judul

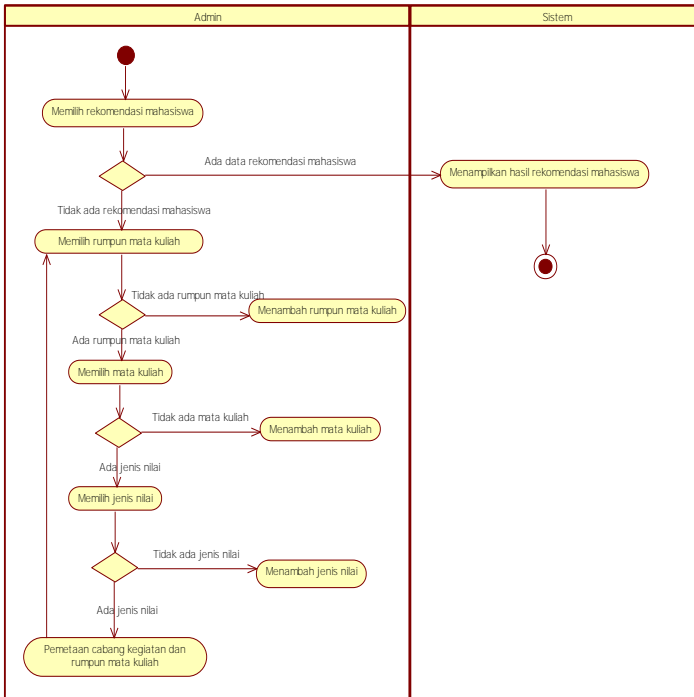
	A1.3 Mahasiswa memasukan isi berita A2. Mahasiswa memilih pilihan hapus berita A2.1 Mahasiswa memilih hapus berita A2.2 Mahasiswa menyetujui hapus berita
Kondisi Akhir	Berita berhasil ditambahkan dalam <i>database</i> atau berhasil diubah.



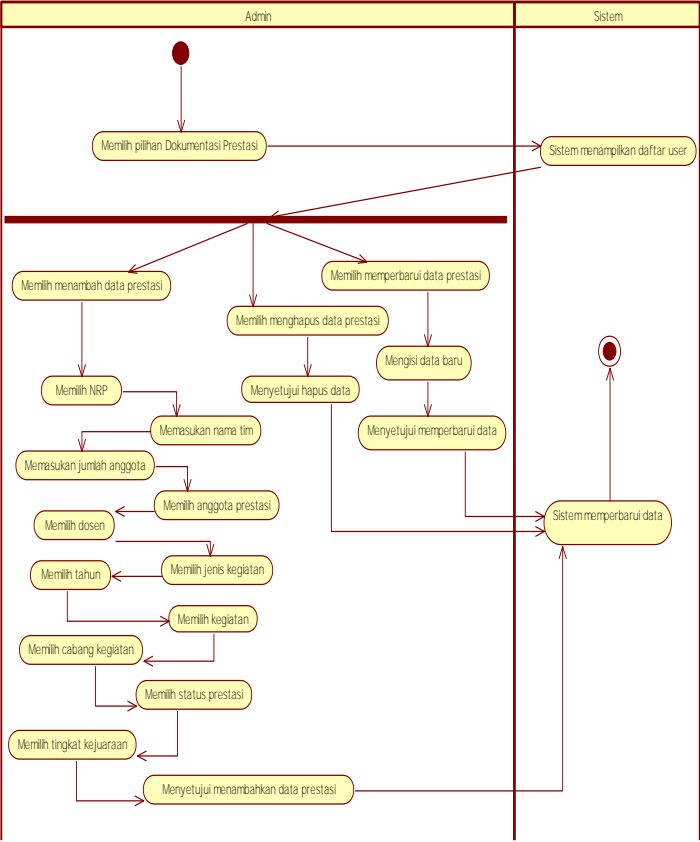
Gambar 8.2 Diagram Aktivitas Mengelola Data Cabang Kegiatan



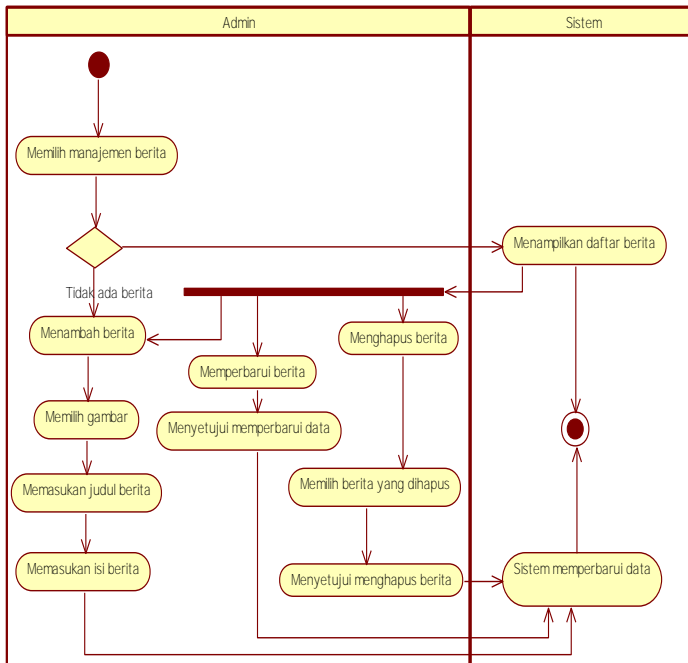
Gambar 8.3 Diagram Aktivitas Kasus Penggunaan Mengelola Permohonan Dana



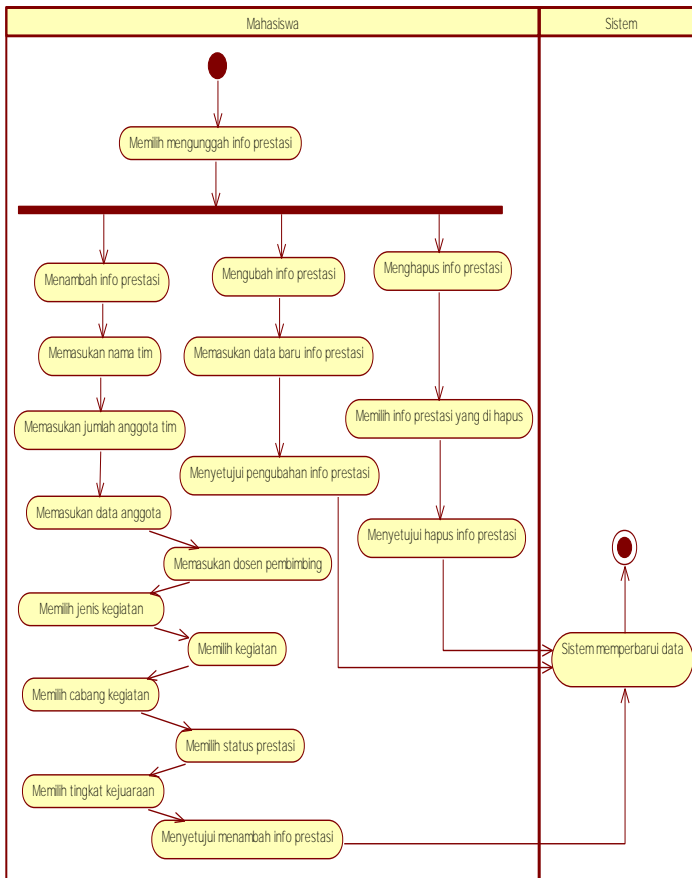
Gambar 8.4 Diagram Aktivitas Kasus Penggunaan Melihat Rekomendasi Mahasiswa



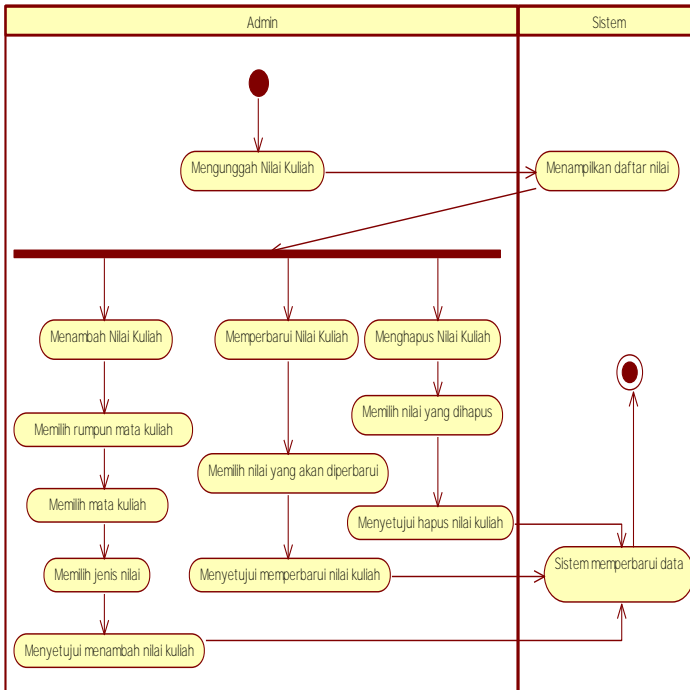
Gambar 8.5 Diagram Aktivitas Kasus Penggunaan Melihat Rekomendasi Prestasi



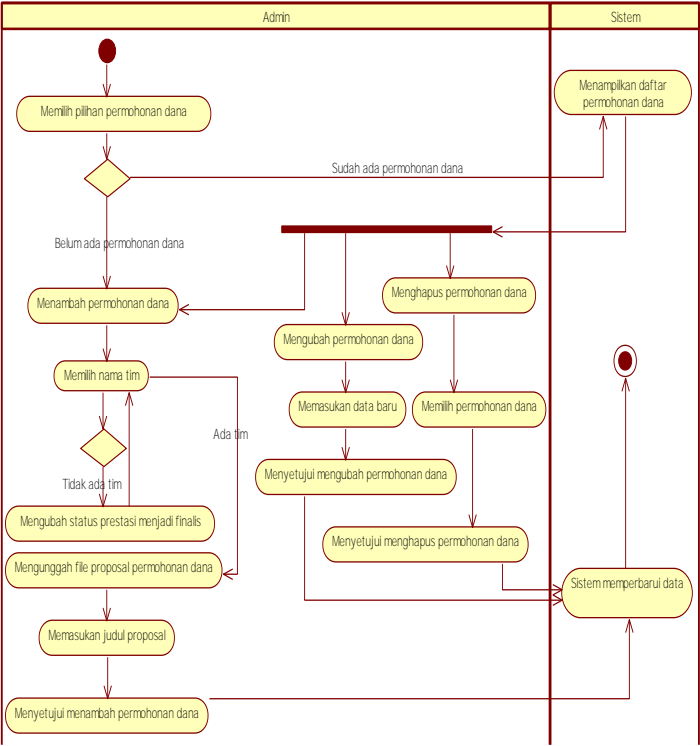
Gambar 8.6 Diagram Aktivitas Kasus Penggunaan Mengelola Berita



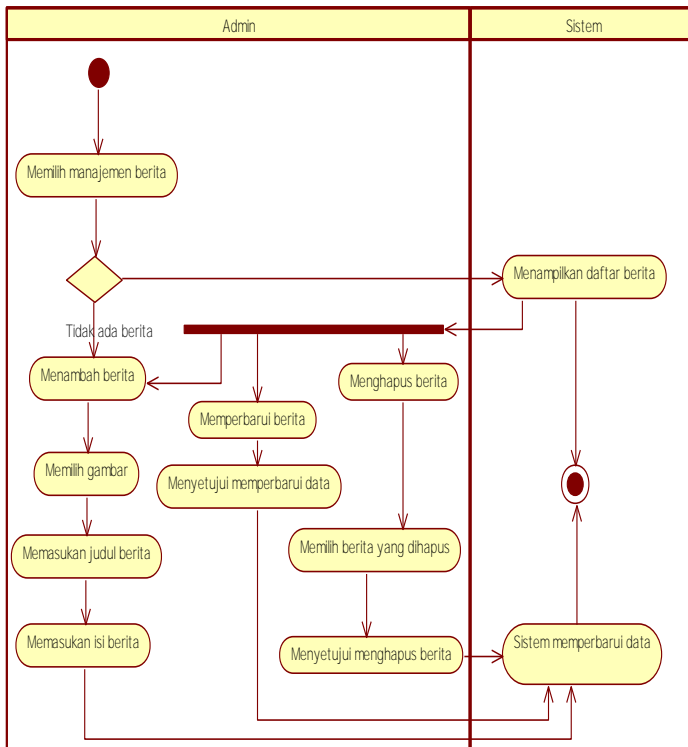
Gambar 8.7 Diagram Aktivitas Kasus Penggunaan Mengunggah Info Prestasi



Gambar 8.8 Diagram Aktivitas Kasus Penggunaan Mengunggah Nilai Kuliah

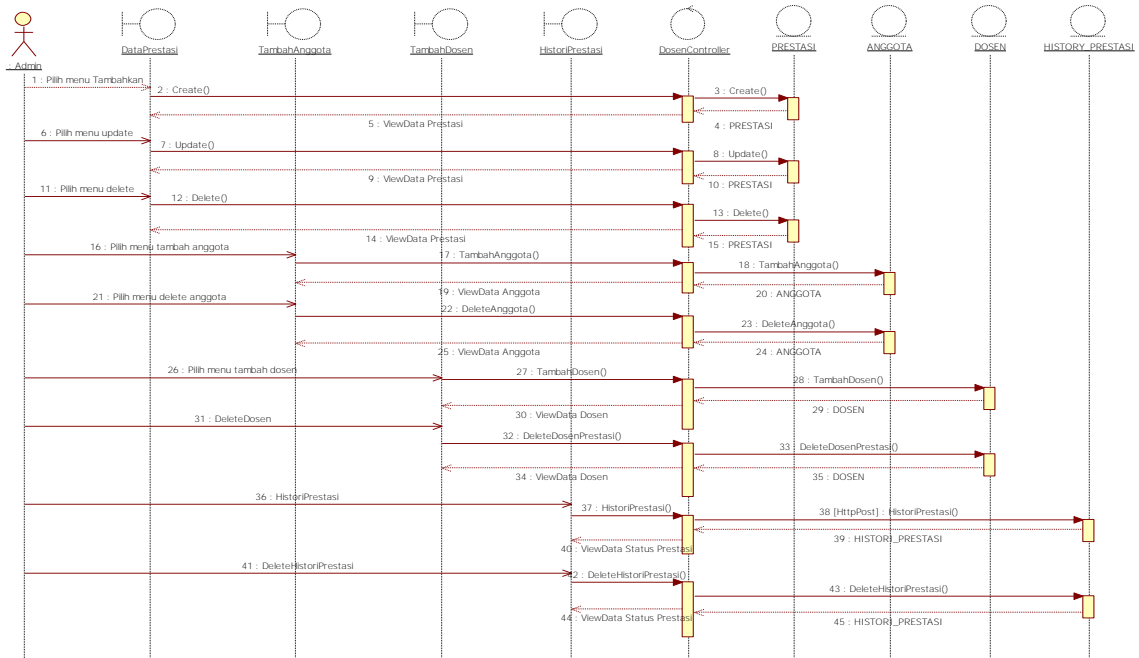


Gambar 8.9 Diagram Aktivitas Kasus Penggunaan Mengunggah Permohonan Dana

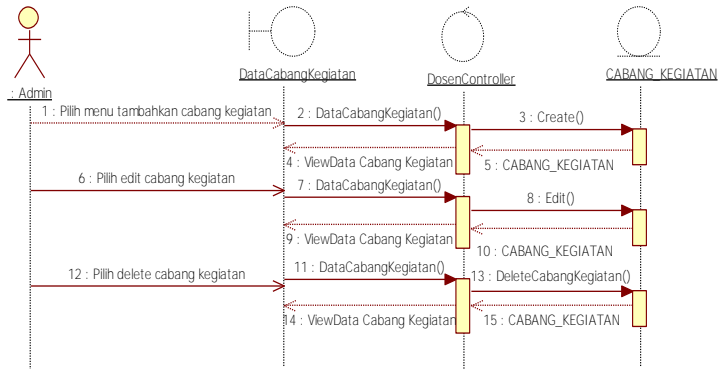


Gambar 8.10 Diagram Aktivitas Kasus Penggunaan

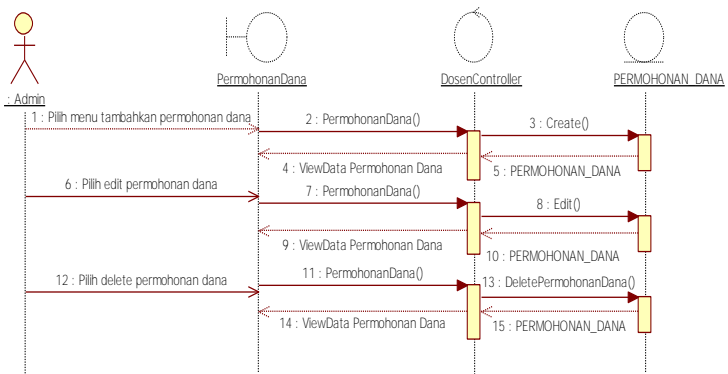
8.3. Diagram Sekuens



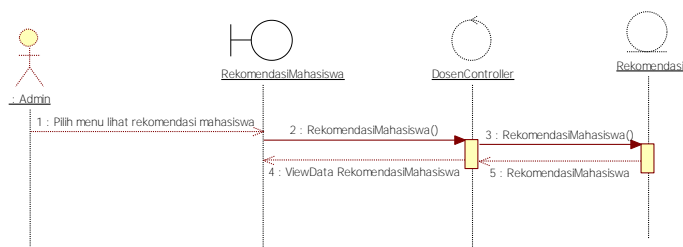
Gambar 8.11 Diagram Sekuens Mengelola Dokumentasi Prestasi



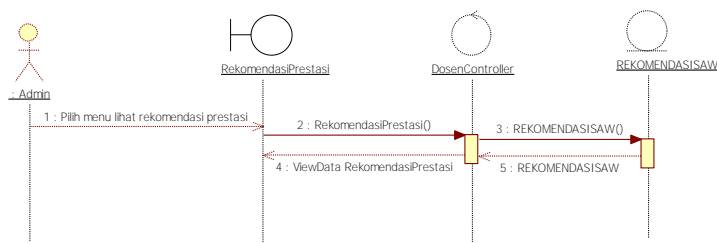
Gambar 8.12 Diagram Sekuens Mengelola Cabang Kegiatan



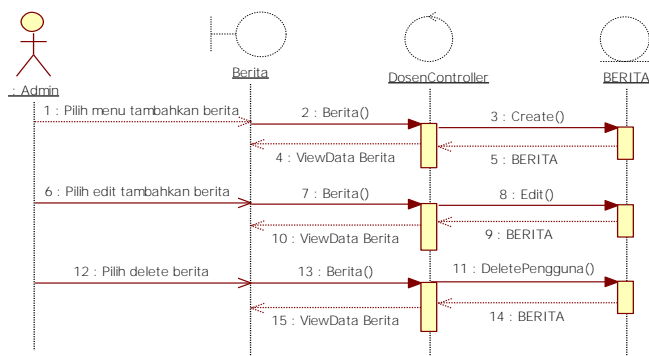
Gambar 8.13 Diagram Sekuens Mengelola Permohonan Dana



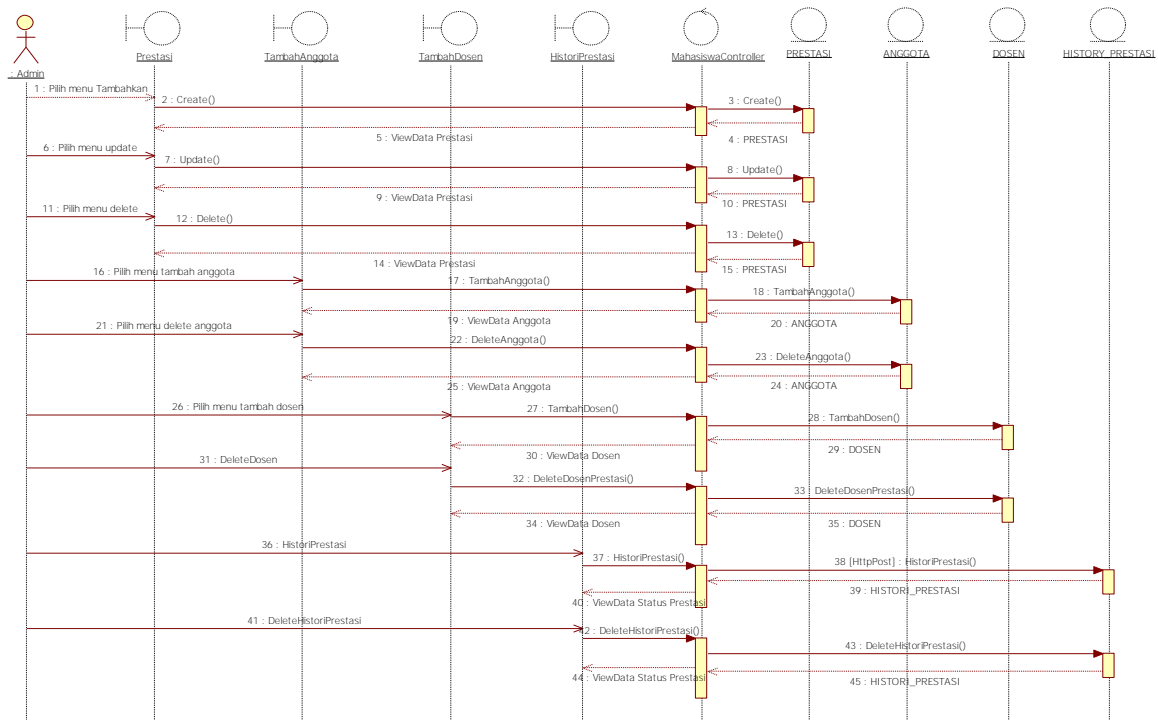
Gambar 8.14 Diagram Sekuens Rekomendasi Mahasiswa



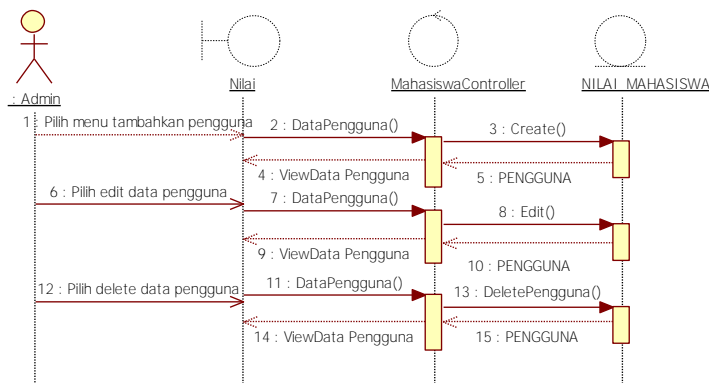
Gambar 8.15 Diagram Sekuens Rekomendasi Prestasi



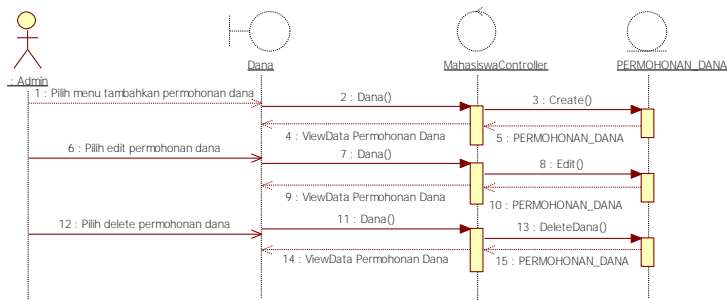
Gambar 8.16 Diagram Sekuens Mengelola Berita



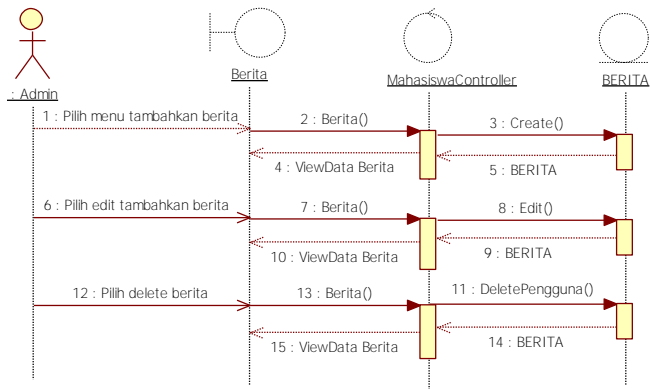
Gambar 8.17 Diagram Sekuens Mengunggah Info Prestasi



Gambar 8.18 Diagram Sekuens Mengunggah Nilai



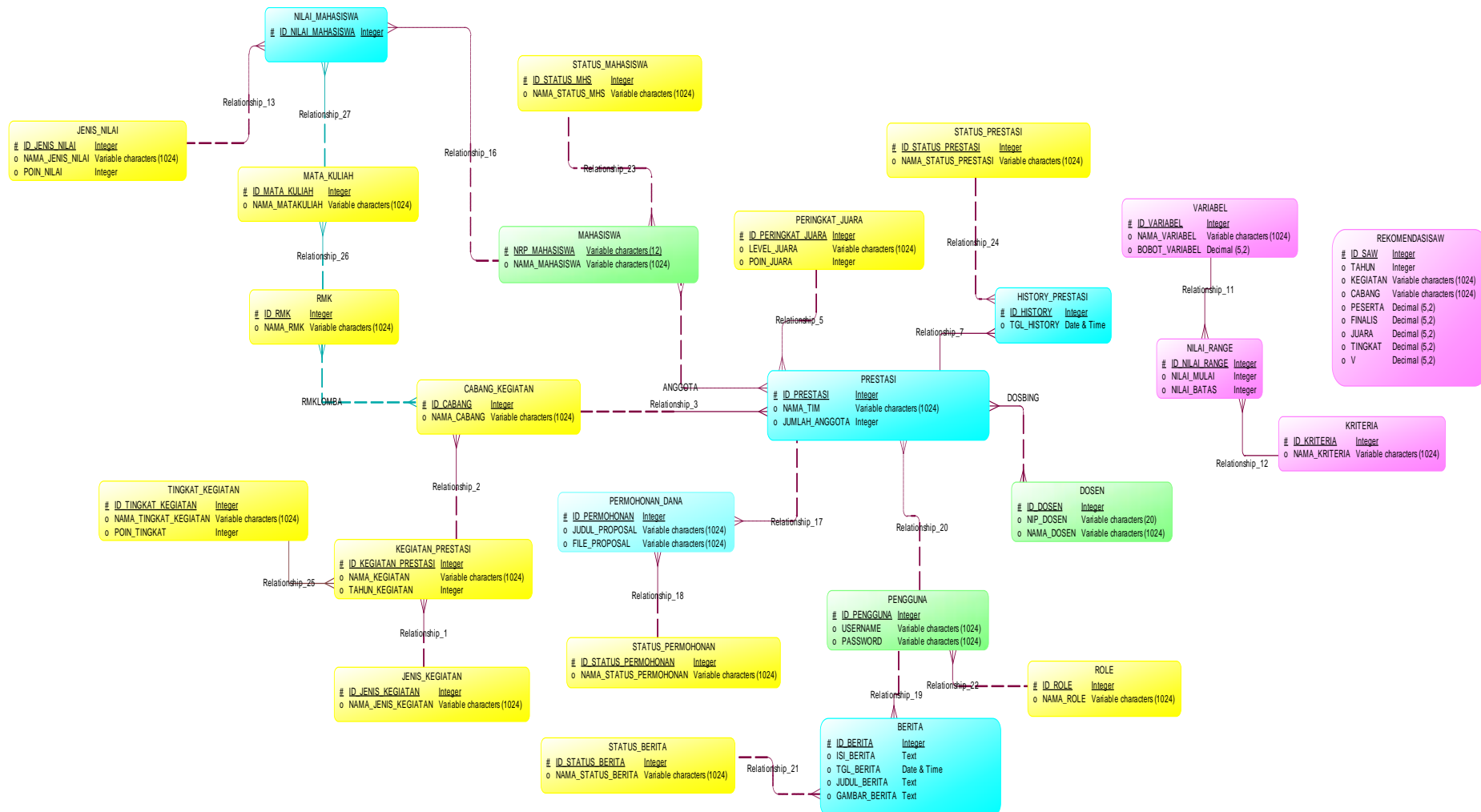
Gambar 8.19 Diagram Sekuens Mengunggah Permohonan Dana



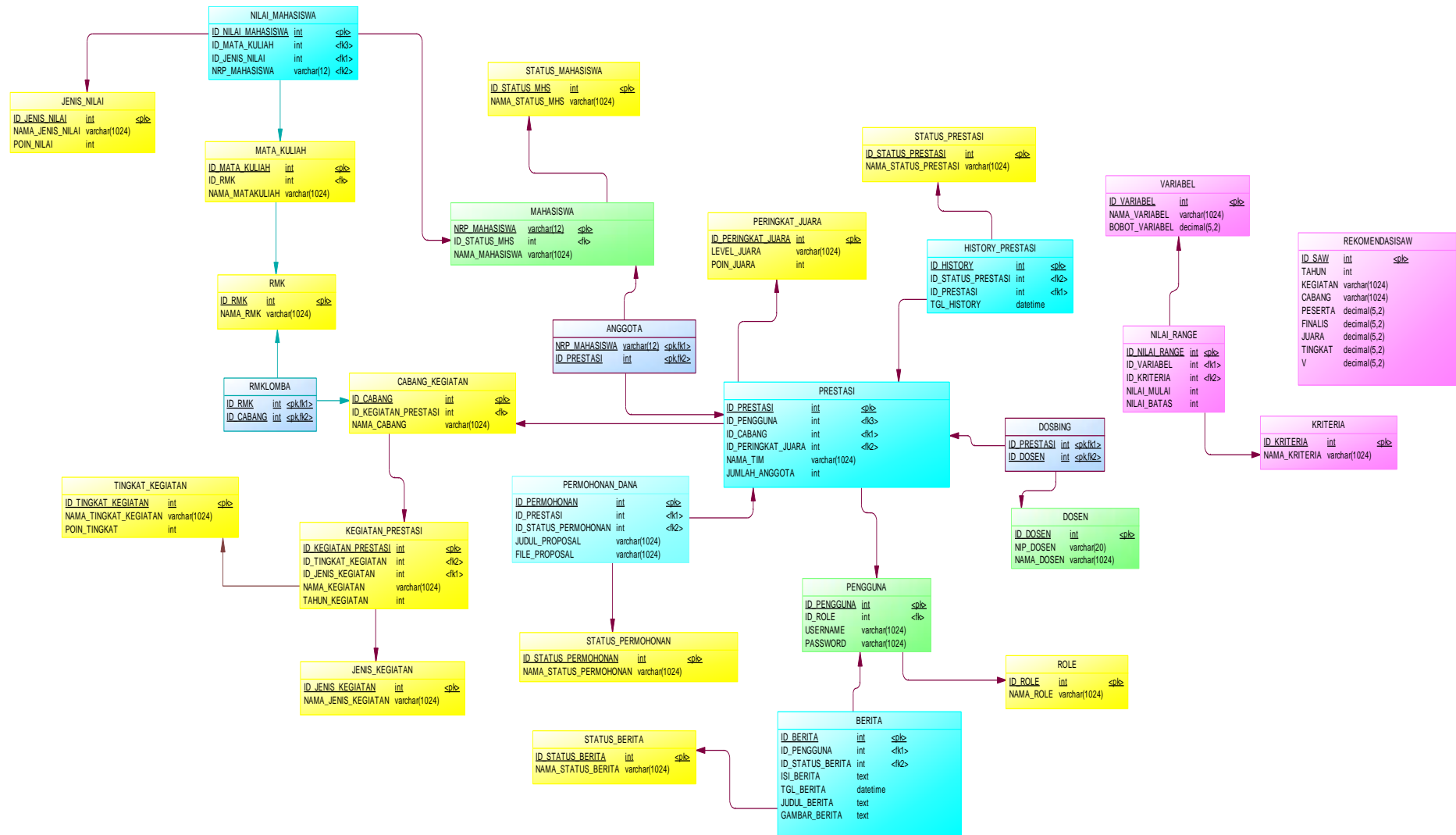
Gambar 8.20 Diagram Sekuens Mengunggah Berita

[Halaman ini sengaja dikosongkan]

8.4. Perancangan Database

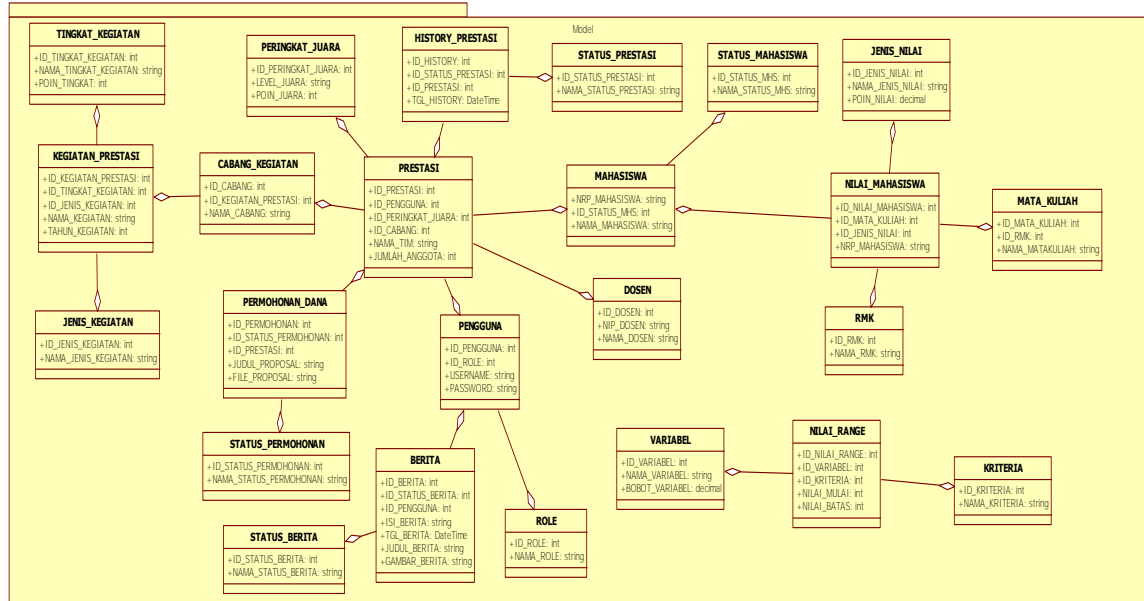


Gambar 8.21 Conceptual Data Model Aplikasi Juara

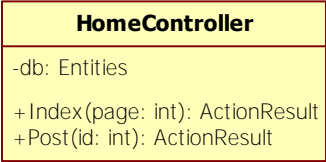


Gambar 8.22 Physical Data Model Aplikasi Juara

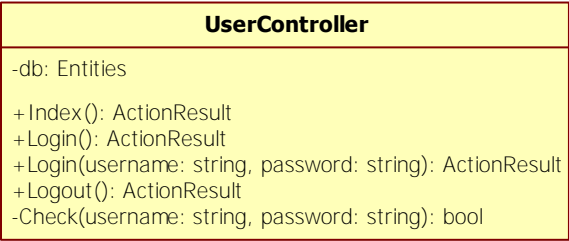
8.5. Perancangan Diagram Kelas



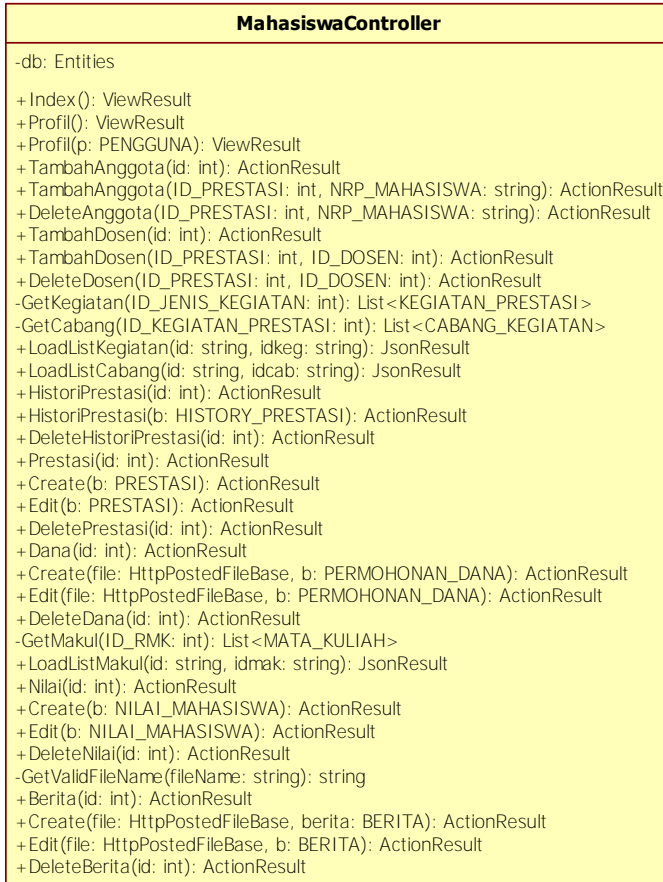
Gambar 8.23 Diagram Kelas Model Aplikasi JUARA



Gambar 8.24 Diagram Kelas Kontrol Home Controller



Gambar 8.25 Diagram Kelas Kontrol UserController



Gambar 8.26 Diagram Kelas Kontrol MahasiswaController

DosenController
-db: Entities
+Index(): ActionResult +UpdateRekomendasiPrestasiIndex(): ActionResult +Profil(): ViewResult +Profil(p: PENGGUNA): ViewResult +DataPengguna(id: int): ActionResult +Create(b: PENGGUNA): ActionResult +Edit(b: PENGGUNA): ActionResult +DeletePengguna(id: int): ActionResult +DataMahasiswa(id: string): ActionResult +Create(b: MAHASISWA): ActionResult +Edit(b: MAHASISWA): ActionResult +DeleteMahasiswa(id: string): ActionResult +DataDosen(id: int): ActionResult +Create(b: DOSEN): ActionResult +Edit(b: DOSEN): ActionResult +DeleteDosen(id: int): ActionResult -GetKegiatan(ID_JENIS_KEGIATAN: int): List<KEGIATAN_PRESTASI> -GetCabang(ID_KEGIATAN_PRESTASI: int): List<CABANG_KEGIATAN> +LoadListKegiatan(id: string, idkeg: string): JsonResult +LoadListCabang(id: string, idcab: string): JsonResult +TambahAnggota(id: int): ActionResult +TambahAnggota(ID_PRESTASI: int, NRP_MAHASISWA: string): ActionResult +DeleteAnggota(ID_PRESTASI: int, NRP_MAHASISWA: string): ActionResult +TambahDosen(id: int): ActionResult +TambahDosen(ID_PRESTASI: int, ID_DOSEN: int): ActionResult +DeleteDosenPrestasi(ID_PRESTASI: int, ID_DOSEN: int): ActionResult +HistoriPrestasi(id: int): ActionResult +HistoriPrestasi(b: HISTORY_PRESTASI): ActionResult +DeleteHistoriPrestasi(id: int): ActionResult +DataPrestasi(id: int): ActionResult +Create(b: PRESTASI): ActionResult

Gambar 8.27 Diagram Kelas Kontrol DosenController

Gambar 8.28 Diagram Kelas Aplikasi JUARA

[Halaman ini sengaja dikosongkan]

LAMPIRAN B. KODE SUMBER

```
public ActionResult UpdateRekomendasiPrestasiIndex()
{
    TypeBuilder builder =
DosenController.CreateTypeBuilder("MyDynamicAssembly",
"MyModule", "MyType");

DosenController.CreateAutoImplementedProperty(builder,
"TAHUN_KEGIATAN", typeof(int));

DosenController.CreateAutoImplementedProperty(builder,
"NAMA_KEGIATAN", typeof(string));

DosenController.CreateAutoImplementedProperty(builder,
"NAMA_CABANG", typeof(string));

DosenController.CreateAutoImplementedProperty(builder,
"PESERTA", typeof(decimal));

DosenController.CreateAutoImplementedProperty(builder,
"FINALIS", typeof(decimal));

DosenController.CreateAutoImplementedProperty(builder,
"POIN_JUARA", typeof(decimal));

DosenController.CreateAutoImplementedProperty(builder,
"POIN_TINGKAT", typeof(decimal));

    Type resultType = builder.CreateType();

    dynamic queryResult =
db.Database.SqlQuery(resultType, "SELECT TA.TAHUN_KEGIATAN,
TA.NAMA_KEGIATAN, TA.NAMA_CABANG, SUM(TA.PESERTA) AS PESERTA,
SUM(TA.FINALIS) AS FINALIS, SUM(TA.POIN_JUARA) AS POIN_JUARA,
SUM(TA.POIN_TINGKAT) AS POIN_TINGKAT FROM (select
P.NAMA_KEGIATAN, P.NAMA_CABANG, P.TAHUN_KEGIATAN,
    PESERTA=CASE N.ID_KRITERIA WHEN 1 THEN ( CASE WHEN
P.peserta BETWEEN N.NILAI_MULAI AND N.NILAI_BATAS THEN
    N.BOBOT_VARIABEL ELSE 0 ) ELSE 0 END,
FINALIS=CASE N.ID_KRITERIA WHEN 2 THEN ( CASE WHEN
P.finalis BETWEEN N.NILAI_MULAI AND N.NILAI_BATAS THEN
    N.BOBOT_VARIABEL ELSE 0 ) ELSE 0 END,
POIN_JUARA =CASE N.ID_KRITERIA WHEN 3 THEN ( CASE WHEN
P.poin_juara BETWEEN N.NILAI_MULAI AND N.NILAI_BATAS THEN
```

```

        N.BOBOT_VARIABEL          ELSE 0  END ) ELSE 0 END,
    POIN_TINGKAT=CASE N.ID_KRITERIA WHEN 4 THEN ( CASE WHEN
P.poin_tingkat BETWEEN N.NILAI_MULAI AND N.NILAI_BATAS THEN
        N.BOBOT_VARIABEL          ELSE 0  END ) ELSE 0 END from
( select a.TAHUN_KEGIATAN, a.NAMA_KEGIATAN,a.NAMA_CABANG,
SUM(a.PESERTA) as peserta, SUM(a.FINALIS) as finalis,
SUM(a.POIN_JUARA) as poin_juara, a.POIN_TINGKAT as
poin_tingkat from ( SELECT
KEGIATAN_PRESTASI.TAHUN_KEGIATAN,KEGIATAN_PRESTASI.NAMA_KEGIAT
AN,CABANG_KEGIATAN.NAMA_CABANG,
        PESERTA=CASE STATUS_PRESTASI.NAMA_STATUS_PRESTASI
                WHEN 'PESERTA' THEN
SUM(PRESTASI.JUMLAH_ANGGOTA)          ELSE 0
                END,          FINALIS=CASE
STATUS_PRESTASI.NAMA_STATUS_PRESTASI          WHEN
'FINALIS' THEN SUM(PRESTASI.JUMLAH_ANGGOTA)
                ELSE 0          END,
        POIN_JUARA=CASE STATUS_PRESTASI.NAMA_STATUS_PRESTASI
                WHEN 'JUARA' THEN
SUM(PERINGKAT_JUARA.POIN_JUARA)          ELSE 0
                END,
        TINGKAT_KEGIATAN.POIN_TINGKAT          from
                PRESTASI
                LEFT JOIN HISTORY_PRESTASI ON
(PRESTASI.ID_PRESTASI=HISTORY_PRESTASI.ID_PRESTASI)
                LEFT JOIN STATUS_PRESTASI ON
(HISTORY_PRESTASI.ID_STATUS_PRESTASI=STATUS_PRESTASI.ID_STATUS
_PRESTASI)
                LEFT JOIN CABANG_KEGIATAN ON
(PRESTASI.ID_CABANG=CABANG_KEGIATAN.ID_CABANG)
                LEFT JOIN KEGIATAN_PRESTASI ON
(CABANG_KEGIATAN.ID_KEGIATAN_PRESTASI=KEGIATAN_PRESTASI.ID_KEG
IATAN_PRESTASI)
                LEFT JOIN PERINGKAT_JUARA ON
(PRESTASI.ID_PERINGKAT_JUARA=PERINGKAT_JUARA.ID_PERINGKAT_JUAR
A)
                LEFT JOIN TINGKAT_KEGIATAN ON
(KEGIATAN_PRESTASI.ID_TINGKAT_KEGIATAN=TINGKAT_KEGIATAN.ID_TIN
GKAT_KEGIATAN) group by KEGIATAN_PRESTASI.NAMA_KEGIATAN,
CABANG_KEGIATAN.NAMA_CABANG, KEGIATAN_PRESTASI.TAHUN_KEGIATAN,
STATUS_PRESTASI.NAMA_STATUS_PRESTASI,
TINGKAT_KEGIATAN.POIN_TINGKAT ) as a group by a.NAMA_KEGIATAN,
a.NAMA_CABANG, a.TAHUN_KEGIATAN, a.POIN_TINGKAT )P, (SELECT
NILAI_RANGE.ID_KRITERIA, NILAI_RANGE.ID_VARIABEL,
KRITERIA.NAMA_KRITERIA, NILAI_RANGE.NILAI_MULAI,
NILAI_RANGE.NILAI_BATAS, VARIABEL.NAMA_VARIABEL,
VARIABEL.BOBOT_VARIABEL FROM KRITERIA INNER JOIN NILAI_RANGE
ON KRITERIA.ID_KRITERIA = NILAI_RANGE.ID_KRITERIA INNER JOIN
VARIABEL ON NILAI_RANGE.ID_VARIABEL = VARIABEL.ID_VARIABEL)
N ) TA GROUP BY TA.NAMA_KEGIATAN, TA.NAMA_CABANG,

```



```

TA.TAHUN_KEGIATAN");
    decimal maxpeserta = 0m;
    decimal maxfinalis = 0m;
    decimal maxjuara = 0m;
    decimal maxtingkat = 0m;
    foreach (dynamic item in queryResult)
    {
        if (item.PESERTA >= maxpeserta)
maxpeserta = item.PESERTA;
        if (item.FINALIS >= maxfinalis) maxfinalis =
item.FINALIS;
        if (item.POIN_JUARA >= maxjuara) maxjuara =
item.POIN_JUARA;
        if (item.POIN_TINGKAT >= maxtingkat)
maxtingkat = item.POIN_TINGKAT;
    }
    REKOMENDASISAW W = new REKOMENDASISAW()
    {
        PESERTA = .2m,
        FINALIS = .8m,
        JUARA = 1m,
        TINGKAT = .6m,
    };
    db.REKOMENDASISAWs.ToList().ForEach(x =>
db.REKOMENDASISAWs.Remove(x));
    List<REKOMENDASISAW> list = new
List<REKOMENDASISAW>();
    foreach (dynamic item in queryResult)
    {
        list.Add(new REKOMENDASISAW()
        {
            TAHUN = item.TAHUN_KEGIATAN,
            KEGIATAN = item.NAMA_KEGIATAN,
            CABANG = item.NAMA_CABANG,
            PESERTA = item.PESERTA / maxpeserta,
            FINALIS = item.FINALIS / maxfinalis,
            JUARA = item.POIN_JUARA / maxjuara,
            TINGKAT = item.POIN_TINGKAT / maxtingkat,
        });
    }
    foreach (REKOMENDASISAW item in list)
    {
        item.V = W.PESERTA * item.PESERTA + W.FINALIS
* item.FINALIS + W.JUARA * item.JUARA +
W.TINGKAT * item.TINGKAT;
    }
    list = list.OrderByDescending(x => x.V).ToList();

```

```

        int rank = 1;
        foreach (REKOMENDASISAW item in list)
        {
            item.ID_SAW = rank++;
            item.TGL_UPDATE = DateTime.Now;
            db.REKOMENDASISAWs.Add(item);
        }
        if (ModelState.IsValid)
        {
            db.SaveChanges();
        }
        return RedirectToAction("Index");
    }
}

```

Kode Sumber 9.1 Update Rekomendasi Prestasi

```

#region Profil
public ActionResult Profil()
{
    PENGGUNA p =
db.PENGGUNAs.Find(Statis.GetID(((Sesi)Session["user"]).UserNam
e));

    return View(p);
}

[HttpPost]
public ActionResult Profil(PENGGUNA p)
{
    PENGGUNA a =
db.PENGGUNAs.Find(Statis.GetID(((Sesi)Session["user"]).UserNam
e));

    a.PASSWORD = p.PASSWORD;
    if (ModelState.IsValid)
    {
        db.Entry(a).State = EntityState.Modified;
        db.SaveChanges();
    }
    ModelState.AddModelError("", "Password baru
tersimpan.");
    return View(p);
}
#endregion

```

Kode Sumber 9.2 Region Profil

```

#region DataPengguna
public ActionResult DataPengguna(int? id)

```

```

    {
        ViewData["Operation"] = id;

        ViewData["list"] = db.PENGGUNAS.ToList();
        PENGGUNA b = db.PENGGUNAS.Find(id);
        ViewBag.ID_ROLE = new SelectList(db.ROLES,
        "ID_ROLE", "NAMA_ROLE", (b != null) ? b.ID_ROLE : 0);
        return View(b);
    }

    [HttpPost]
    [ActionName("DataPengguna")]
    [OnAction(ButtonName = "Create")]
    public ActionResult Create(PENGGUNA b)
    {
        if (ModelState.IsValid)
        {
            db.PENGGUNAS.Add(b);
            db.SaveChanges();
        }
        return RedirectToAction("DataPengguna");
    }

    [HttpPost]
    [ActionName("DataPengguna")]
    [OnAction(ButtonName = "Update")]
    public ActionResult Edit(PENGGUNA b)
    {
        if (ModelState.IsValid)
        {
            db.Entry(b).State = EntityState.Modified;
            db.SaveChanges();
        }
        return RedirectToAction("DataPengguna", new { id =
0 });
    }

    public ActionResult DeletePengguna(int id)
    {
        PENGGUNA p = db.PENGGUNAS.Find(id);
        db.PENGGUNAS.Remove(p);
        db.SaveChanges();
        return RedirectToAction("DataPengguna", new { id =
0 });
    }

```

```
}

```

```
#endregion

```

Kode Sumber 9.3 Region Data Pengguna

```
#region DataPrestasi

    // ===== prestasi anggota
    dosen

    private IList<KEGIATAN_PRESTASI> GetKegiatan(int
ID_JENIS_KEGIATAN, int tahun)
    {
        return db.KEGIATAN_PRESTASI.Where(m =>
m.ID_JENIS_KEGIATAN == ID_JENIS_KEGIATAN && m.TAHUN_KEGIATAN
== tahun).ToList();
    }

    private IList<CABANG_KEGIATAN> GetCabang(int
ID_KEGIATAN_PRESTASI)
    {
        return db.CABANG_KEGIATAN.Where(m =>
m.ID_KEGIATAN_PRESTASI == ID_KEGIATAN_PRESTASI).ToList();
    }

    [AcceptVerbs(HttpVerbs.Get)]
    public JsonResult LoadListTahun2(int? tahun)
    {
        var classesList = db.KEGIATAN_PRESTASI.GroupBy(x
=> x.TAHUN_KEGIATAN);
        var classesData = classesList.ToList().Select(m =>
new SelectListItem()
        {
            Selected = (m.Key.Value ==
Convert.ToInt32(tahun)),
            Text = m.Key.Value.ToString(),
            Value = m.Key.Value.ToString(),
        });
        return Json(classesData,
JsonRequestBehavior.AllowGet);
    }

    [AcceptVerbs(HttpVerbs.Get)]
    public JsonResult LoadListKegiatan(string id, string
idkeg, string tahun)
    {

```

```

        var classesList =
this.GetKegiatan(Convert.ToInt32(id), Convert.ToInt32(tahun));
        var classesData = classesList.Select(m => new
SelectListItem()
        {
            Selected = (m.ID_KEGIATAN_PRESTASI ==
Convert.ToInt64(idkeg)),
            Text = m.NAMA_KEGIATAN,
            Value = m.ID_KEGIATAN_PRESTASI.ToString(),
        });
        return Json(classesData,
JsonRequestBehavior.AllowGet);
    }

    [AcceptVerbs(HttpVerbs.Get)]
    public JsonResult LoadListCabang(string id, string
idcab)
    {
        var classesList =
this.GetCabang(Convert.ToInt32(id));
        var classesData = classesList.Select(m => new
SelectListItem()
        {
            Selected = (m.ID_CABANG ==
Convert.ToInt64(idcab)),
            Text = m.NAMA_CABANG,
            Value = m.ID_CABANG.ToString(),
        });
        return Json(classesData,
JsonRequestBehavior.AllowGet);
    }

    public ActionResult TambahAnggota(int? id)
    {
        PRESTASI b = db.PRESTASIs.Find(id);
        if (b == null) return RedirectToAction("Prestasi",
new { id = 0 });
        ViewData["list"] = b.MAHASISWAs.ToList();
        ViewBag.NRP_MAHASISWA = new
SelectList(db.MAHASISWAs, "NRP_MAHASISWA", "NAMA_MAHASISWA");
        return View(b);
    }

    [HttpPost]
    public ActionResult TambahAnggota(int? ID_PRESTASI,
string NRP_MAHASISWA = null)
    {
        PRESTASI b = db.PRESTASIs.Find(ID_PRESTASI);

```

```

        if (b.MAHASISWAs.Count >= b.JUMLAH_ANGGOTA)
        {
            ModelState.AddModelError("", "Jumlah anggota
telah memenuhi kuota (" + b.JUMLAH_ANGGOTA + " anggota).");
            ViewBag.NRP_MAHASISWA = new
SelectList(db.MAHASISWAs, "NRP_MAHASISWA", "NAMA_MAHASISWA");
            return View(b);
        }
        MAHASISWA p = db.MAHASISWAs.Find(NRP_MAHASISWA);
        b.MAHASISWAs.Add(p);
        db.Entry(b).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("TambahAnggota", new { id
= ID_PRESTASI });
    }

    public ActionResult DeleteAnggota(int? ID_PRESTASI,
string NRP_MAHASISWA = null)
    {
        PRESTASI b = db.PRESTASIs.Find(ID_PRESTASI);
        MAHASISWA p = db.MAHASISWAs.Find(NRP_MAHASISWA);
        b.MAHASISWAs.Remove(p);
        db.Entry(b).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("TambahAnggota", new { id
= ID_PRESTASI });
    }

    public ActionResult TambahDosen(int? id)
    {
        PRESTASI b = db.PRESTASIs.Find(id);
        if (b == null) return RedirectToAction("Prestasi",
new { id = 0 });
        ViewData["list"] = b.DOSENs.ToList();
        ViewBag.ID_DOSEN = new SelectList(db.DOSENs,
"ID_DOSEN", "NAMA_DOSEN");
        return View(b);
    }

    [HttpPost]
    public ActionResult TambahDosen(int? ID_PRESTASI, int?
ID_DOSEN)
    {
        PRESTASI b = db.PRESTASIs.Find(ID_PRESTASI);
        DOSEN p = db.DOSENs.Find(ID_DOSEN);
        b.DOSENs.Add(p);
        db.Entry(b).State = EntityState.Modified;
        db.SaveChanges();
    }

```

```

        return RedirectToAction("TambahDosen", new { id =
ID_PRESTASI });

    }

    public ActionResult DeleteDosenPrestasi(int?
ID_PRESTASI, int? ID_DOSEN)
    {
        PRESTASI b = db.PRESTASIs.Find(ID_PRESTASI);
        DOSEN p = db.DOSENs.Find(ID_DOSEN);
        b.DOSENs.Remove(p);
        db.Entry(b).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("TambahDosen", new { id =
ID_PRESTASI });
    }
    // ===== prestasi anggota
dosen

    public ActionResult HistoriPrestasi(int? id)
    {
        PRESTASI p = db.PRESTASIs.Find(id);
        ViewData["list"] = p.HISTORY_PRESTASI.ToList();
        int min = (p.HISTORY_PRESTASI.Count > 0) ?
p.HISTORY_PRESTASI.LastOrDefault().STATUS_PRESTASI.ID_STATUS_P
RESTASI : 0;
        ViewBag.ID_STATUS_PRESTASI = new
SelectList(db.STATUS_PRESTASI.Where(x => x.ID_STATUS_PRESTASI
> min), "ID_STATUS_PRESTASI", "NAMA_STATUS_PRESTASI");
        HISTORY_PRESTASI b = new HISTORY_PRESTASI();
        b.ID_PRESTASI = p.ID_PRESTASI;
        return View(b);
    }

    [HttpPost]
    public ActionResult HistoriPrestasi(HISTORY_PRESTASI
b)
    {
        b.TGL_HISTORY = DateTime.Now;
        if (ModelState.IsValid)
        {
            db.HISTORY_PRESTASI.Add(b);
            db.SaveChanges();
        }
        return RedirectToAction("HistoriPrestasi");
    }

```

```

        public ActionResult DeleteHistoriPrestasi(int id)
        {
            HISTORY_PRESTASI p = db.HISTORY_PRESTASI.Find(id);
            int idpres = p.ID_PRESTASI;
            db.HISTORY_PRESTASI.Remove(p);
            db.SaveChanges();
            return RedirectToAction("HistoriPrestasi", new
{ id = idpres });
        }

        public ActionResult DataPrestasi(int? id)
        {
            ViewData["Operation"] = id;
            ViewData["list"] = db.PRESTASIs.ToList();
            PRESTASI b = db.PRESTASIs.Find(id);
            var listMhs = db.MAHASISWAs.Select(r =>
r.NRP_MAHASISWA);
            var penggunaMhs = db.PENGGUNAs.Where(r =>
listMhs.Contains(r.USERNAME));
            ViewBag.ID_PENGGUNA = new SelectList(penggunaMhs,
"ID_PENGGUNA", "USERNAME", (b != null) ? b.ID_PENGGUNA : 0);
            ViewBag.ID_JENIS_KEGIATAN = new
SelectList(db.JENIS_KEGIATAN, "ID_JENIS_KEGIATAN",
"NAMA_JENIS_KEGIATAN", (b != null) ?
b.CABANG_KEGIATAN.KEGIATAN_PRESTASI.ID_JENIS_KEGIATAN : 0);
            ViewBag.ID_KEGIATAN_PRESTASI = new
SelectList(db.KEGIATAN_PRESTASI, "ID_KEGIATAN_PRESTASI",
"NAMA_KEGIATAN", (b != null) ?
b.CABANG_KEGIATAN.ID_KEGIATAN_PRESTASI : 0);
            ViewBag.ID_CABANG = new
SelectList(db.CABANG_KEGIATAN, "ID_CABANG", "NAMA_CABANG",
(b != null) ? b.ID_CABANG : 0);
            ViewBag.ID_PERINGKAT_JUARA = new
SelectList(db.PERINGKAT_JUARA, "ID_PERINGKAT_JUARA",
"LEVEL_JUARA", (b != null) ? b.ID_PERINGKAT_JUARA : 0);
            return View(b);
        }
        [HttpPost]
        [ActionName("DataPrestasi")]
        [OnAction(ButtonName = "Create")]
        public ActionResult Create(PRESTASI b)
        {
            PENGGUNA g = db.PENGGUNAs.Find(b.ID_PENGGUNA);
            MAHASISWA p = db.MAHASISWAs.Find(g.USERNAME);
            b.MAHASISWAs.Add(p);
            b.ID_PERINGKAT_JUARA = 6;
        }
    }

```



```

        if (ModelState.IsValid)
        {

            db.PRESTASIs.Add(b);
            db.SaveChanges();
            HISTORY_PRESTASI h = new HISTORY_PRESTASI();
            h.ID_PRESTASI = b.ID_PRESTASI;
            h.ID_STATUS_PRESTASI = 1;
            h.TGL_HISTORY = DateTime.Now;
            db.HISTORY_PRESTASI.Add(h);
            db.SaveChanges();
        }
        return RedirectToAction("DataPrestasi");
    }

    [HttpPost]
    [ActionName("DataPrestasi")]
    [OnAction(ButtonName = "Update")]
    public ActionResult Edit(PRESTASI b)
    {
        PRESTASI old = db.PRESTASIs.Find(b.ID_PRESTASI);
        old.NAMA_TIM = b.NAMA_TIM;
        old.JUMLAH_ANGGOTA = b.JUMLAH_ANGGOTA;
        old.ID_CABANG = b.ID_CABANG;
        if (b.ID_PERINGKAT_JUARA == 0)
        b.ID_PERINGKAT_JUARA = 6;
        old.ID_PERINGKAT_JUARA = b.ID_PERINGKAT_JUARA;
        if (ModelState.IsValid)
        {
            db.Entry(old).State = EntityState.Modified;
            db.SaveChanges();
        }
        return RedirectToAction("DataPrestasi", new { id =
0 });
    }

    public ActionResult DeletePrestasi(int id)
    {
        PRESTASI p = db.PRESTASIs.Find(id);
        p.MAHASISWAs.Clear();

        p.DOSENs.Clear();
        foreach (var r in db.HISTORY_PRESTASI.Where(f =>
f.ID_PRESTASI == id))
        {

```

```

        db.HISTORY_PRESTASI.Remove(r);
    }

    foreach (var r in db.PERMOHONAN_DANA.Where(f =>
f.ID_PRESTASI == id))
    {
        db.PERMOHONAN_DANA.Remove(r);
    }
    db.PRESTASIs.Remove(p);
    db.SaveChanges();
    return RedirectToAction("DataPrestasi", new { id =
0 });
}

#endregion

```

Kode Sumber 9.4 Region Data Prestasi

```

#region Pemetaan

public ActionResult Pemetaan()
{
    ViewData["list"] = db.CABANG_KEGIATAN.ToList();
    return View();
}

public ActionResult TambahPemetaan(int? id)
{
    CABANG_KEGIATAN b = db.CABANG_KEGIATAN.Find(id);
    if (b == null) return
RedirectToAction("Pemetaan");
    ViewData["list"] = b.RMKs.ToList();
    ViewBag.ID_RMK = new SelectList(db.RMKs, "ID_RMK",
"NAMA_RMK");
    return View(b);
}

[HttpPost]
public ActionResult TambahPemetaan(int? ID_CABANG,
int? ID_RMK)
{
    CABANG_KEGIATAN b =
db.CABANG_KEGIATAN.Find(ID_CABANG);
    RMK r = db.RMKs.Find(ID_RMK);
    b.RMKs.Add(r);
    db.Entry(b).State = EntityState.Modified;
    db.SaveChanges();
}

```

```

        return RedirectToAction("TambahPemetaan", new { id
= ID_CABANG });
    }
    public ActionResult DeletePemetaan(int? ID_CABANG,
int? ID_RMK)
    {
        CABANG_KEGIATAN b =
db.CABANG_KEGIATAN.Find(ID_CABANG);
        RMK r = db.RMKs.Find(ID_RMK);
        b.RMKs.Remove(r);
        db.Entry(b).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("TambahPemetaan", new { id
= ID_CABANG });
    }

#endregion Pemetaan

```

Kode Sumber 9.5 Region Pemetaan

```

#region RekomendasiMahasiswa

public class Rekomendasi
{
    public string NRP { get; set; }
    public string NamaMhs { get; set; }
    public string Kegiatan { get; set; }
    public string Cabang { get; set; }
    public decimal? Poin { get; set; }
}

public ActionResult RekomendasiMahasiswa()
{
    var nilai = (
        from n in db.NILAI_MAHASISWA
        from m in db.MAHASISWAS
        from j in db.JENIS_NILAI
        from k in db.MATA_KULIAH
        from r in db.RMKs
        from l in r.CABANG_KEGIATAN

        from p in db.KEGIATAN_PRESTASI
        where n.NRP_MAHASISWA == m.NRP_MAHASISWA
        && n.ID_JENIS_NILAI == j.ID_JENIS_NILAI
        && n.ID_MATA_KULIAH == k.ID_MATA_KULIAH
        && k.ID_RMK == r.ID_RMK
    )
}

```

```

        && l.ID_KEGIATAN_PRESTASI ==
p.ID_KEGIATAN_PRESTASI

        group n by new { n.NRP_MAHASISWA,
n.MAHASISWA.NAMA_MAHASISWA, l.NAMA_CABANG, p.NAMA_KEGIATAN }
into g

        select new
        {
            NRP = g.Key.NRP_MAHASISWA,
            NamaMhs = g.Key.NAMA_MAHASISWA,
            Kegiatan = g.Key.NAMA_KEGIATAN,
            Cabang = g.Key.NAMA_CABANG,
            Poin = g.Sum(x =>
.JENIS_NILAI.POIN_NILAI),
        }
        ).OrderBy(i => i.Cabang).ThenByDescending(i =>
i.Poin).ToList();
        List<Rekomendasi> list = new List<Rekomendasi>();
        foreach (var item in nilai)
        {
            list.Add(new Rekomendasi()
            {
                Cabang = item.Cabang,
                Kegiatan = item.Kegiatan,
                NamaMhs = item.NamaMhs,
                NRP = item.NRP,
                Poin = item.Poin,
            });
        }
        ViewData["list"] = list;
        return View();
    }
#endregion RekomendasiPrestasi

```

Kode Sumber 9.6 Region Rekomendasi Prestasi

LAMPIRAN C. PENGUJIAN

Kuisoner Pengguna

A. Mahasiswa

Data Prestasi

1. Apakah tampilan halaman untuk memasukan info prestasi menarik dan mudah dipahami?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
2. Apakah proses memasukan info prestasi pada website JUARA mudah untuk dilakukan?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

Permohonan Dana

3. Apakah tampilan halaman untuk mengajukan permohonan dana menarik dan mudah dipahami?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
4. Apakah proses mengajukan permohonan dana pada website mudah untuk dilakukan?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

Nilai

5. Apakah tampilan halaman untuk mengunggah nilai menarik dan mudah dipahami?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
6. Apakah proses mengunggah nilai pada website JUARA mudah dilakukan?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

Berita

7. Apakah tampilan halaman untuk mengunggah berita menarik dan mudah dipahami?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
8. Apakah proses mengunggah berita pada website JUARA mudah dilakukan?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

B. Dosen**Data Prestasi**

1. Apakah tampilan halaman untuk manajemen data prestasi menarik dan mudah dipahami?
 1. Sangat Tidak Setuju

2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
2. Apakah proses melakukan manajemen data prestasi pada website JUARA mudah untuk dilakukan?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
 3. Apakah dokumentasi data prestasi pada website JUARA sudah memenuhi kebutuhan dokumentasi prestasi jurusan Teknik Informatika?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

Permohonan Dana

4. Apakah tampilan halaman untuk manajemen permohonan dana menarik dan mudah dipahami?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
5. Apakah proses melakukan manajemen permohonan dana pada website mudah untuk dilakukan?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

6. Apakah proses permohonan dana pada website JUARA sudah sesuai dengan proses bisnis permohonan dana di jurusan Teknik Informatika?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

Rekomendasi Mahasiswa

7. Apakah tampilan halaman untuk manajemen rekomendasi mahasiswa menarik dan mudah dipahami?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
8. Apakah proses melakukan manajemen rekomendasi mahasiswa mudah dipahami?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
9. Apakah hasil rekomendasi mahasiswa bermanfaat untuk mempersiapkan peserta lomba?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

Rekomendasi Prestasi

10. Apakah tampilan halaman untuk manajemen rekomendasi prestasi menarik dan mudah dipahami?
 1. Sangat Tidak Setuju
 2. Tidak Setuju

3. Biasa
 4. Setuju
 5. Sangat Setuju
11. Apakah proses melakukan manajemen rekomendasi prestasi mudah dipahami?
1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
12. Apakah hasil rekomendasi prestasi bermanfaat untuk membantu pemetaan fokus pembinaan perlombaan bagi mahasiswa Teknik Informatika?
1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

Berita

13. Apakah tampilan halaman untuk manajemen berita menarik dan mudah dipahami?
1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
14. Apakah proses melakukan manajemen berita pada website JUARA mudah dilakukan?
1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

C. Umum

1. Apakah desain website JUARA menarik?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju
2. Apakah berita prestasi pada website JUARA informatif?
 1. Sangat Tidak Setuju
 2. Tidak Setuju
 3. Biasa
 4. Setuju
 5. Sangat Setuju

Saran dan kritik untuk website JUARA

.....

.....

.....

.....

.....

Testimoni atau pendapat untuk website JUARA

.....

.....

.....

.....

.....

DAFTAR PERSAMAAN

2.1	10
2.2	10

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Kelas Model STATUS_MAHASISWA ..	61
Kode Sumber 4.2 Region Entities.....	121
Kode Sumber 4.3 Query untuk Bobot untuk Setiap Kriteria	125
Kode Sumber 4.4 ActionResult Create	126
Kode Sumber 4.5 ActionResult Edit.....	126
Kode Sumber 4.6 ActionResult DeletePengguna	127
Kode Sumber 5.1 Query Matrix Kriteria	133
Kode Sumber 9.1 Update Rekomendasi Prestasi	232
Kode Sumber 9.2 Region Profil.....	232
Kode Sumber 9.3 Region Data Pengguna.....	234
Kode Sumber 9.4 Region Data Prestasi	240
Kode Sumber 9.5 Region Pemetaan	241
Kode Sumber 9.6 Region Rekomendasi Prestasi	242

DAFTAR PUSTAKA

- [1] Dikti, *Pedoman Pemilihan Mahasiswa Berprestasi Program Sarjana*. Jakarta: Illah Sailah, 2013.
- [2] Widyanti Deni, Sudana Oka, and Sasmita Arya, "Analysis and Implementation Fuzzy Multi-Attribute Decision Making SAW Method for Selection of High Achieving Students in Faculty Level," *IJCSI International Journal of Computer Science Issues*, vol. 10, no. 1, p. 2, 2013.
- [3] Jose Rolando Guay Paz, "Introducing ASP.NET MVC 4," in *Beginning ASP.NET MVC 4*.: Apress, 2013. [Online]. <http://eclipse.org/>
- [4] BootstrapMaster. [Online]. <https://bootstrapmaster.com/software-house/>
- [5] Adam Freeman, "The MVC Pattern," in *Pro ASP.NET MVC 4 - 4th Edition*.: Apress, 2012.
- [6] What is Entity Framework? [Online]. <http://www.entityframeworktutorial.net/what-is-entityframework.aspx>
- [7] Entity Framework Architecture. [Online]. <http://www.entityframeworktutorial.net/EntityFramework-Architecture.aspx>
- [8] Socialresearchmethods. [Online]. <http://www.socialresearchmethods.net/kb/scallik.php>

BIODATA PENULIS



Penulis bernama Ika Astutik, dengan nama panggilan Ika. Penulis dilahirkan di Nganjuk, 20 September 1991 sebagai anak pertama dari dua bersaudara.

Penulis menempuh pendidikan formal di SDN Sonoageng 1 (1998-2004), SMP Negeri 1 Prambon (2004-2007), SMA Negeri 1 Kediri (2007-2010) dan pendidikan S1 di jurusan Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya.

Di jurusan Teknik Informatika, penulis mengambil rumpun mata kuliah Rekayasa Perangkat Lunak. Penulis memiliki ketertarikan di bidang Software Development Process, Business Analysis, dan Manajemen Informasi. Penulis dalam masa kuliah aktif di berbagai organisasi antara lain Badan Eksekutif Mahasiswa (BEM) ITS dan Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) Sebagai Ketua Departemen Media Informasi (2012-2013). Penulis pernah menjadi asisten dosen Teknik Informatika dengan mata kuliah Analisis dan Perancangan Sistem (APS) dan beberapa mata kuliah di PIKTI-ITS, antara lain Pembuatan Game, Desain Web, Excel untuk Bisnis, Software Perkantoran, dan Pemrograman Android. Penulis dapat dihubungi melalui ika.astutik@gmail.com.